eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Medical Devices with Embedded Electronics: Design and Development Methodology for Start-ups

Ph.D. Thesis

Nerea Arandia Gainza

2023

Supervisors:
Dr. José Ignacio Garate Añibarro
Dr. Jon Mabe Álvarez

Bidea erraztu eta amore ematen utzi ez didazuen guztioi

Familia eta lagunei, eskerrik asko

Abstract

Nowadays, to address the healthcare sector's challenges, biotechnology is continuously demanding innovation. Recent events such as the COVID-19 pandemic, the ageing of the population, the increase in dependency rates and the need to promote personalised healthcare in both hospital and home environments highlight the need to develop more sophisticated, reliable, and connected medical monitoring and diagnostic devices quickly and efficiently.

Embedded connected systems have become a key technology for rapidly developing innovative low-cost diagnostic solutions. Embedded systems can be found in both monitoring and diagnostic systems. Some examples are glucose monitors, pacemakers, wearable monitoring devices, etc.

Being aware of the current opportunity in the sector, an increasing number of biotech start-ups are entering the medical device business. Despite having excellent ideas and technical solutions, many of these start-ups fail. This is due to a lack of knowledge of the healthcare sector and the regulatory requirements that must be met.

The design and development phases of embedded medical devices are regulated by two regulations in Europe: 2017/745, Medical Device Regulation (MDR) and 2017/746, In Vitro Medical Device Regulation (IVDR). Other standards define additional aspects, such as device functional safety (IEC 60601), software lifecycle (IEC 62304), quality management (ISO 13485), risk management (ISO 14971), device usability (IEC 62366) and cybersecurity (IEC 81001-5-1), among others.

A large number of regulatory requirements demand a procedural methodology for performing such developments. After analysing existing product design methodologies, no methodologies have been identified that fully address the needs of start-up companies that wish to develop medical devices.

Therefore, as a main contribution, this thesis presents a methodology that outlines the steps to design and develop embedded

medical devices. As a methodology for start-ups, it aims to minimise the financial investment made during stages with high technological risk. In addition, the evaluation of the device by the client or product manager is encouraged in the early stages of development.

The proposed methodology, which intends to guide start-ups in the medical sector, is validated through its application in the development processes of several medical devices. Many have already obtained the CE marking, demonstrating the methodology's effectiveness. Likewise, the most critical procedures of the methodology are validated by the certification body Société Générale de Surveillance (SGS) within the ISO 13485 certification process conducted in Tekniker, a Research and Development Centre. Finally, studies supporting the proposed methodology's main concepts are reviewed and presented.

Laburpena

Bioteknologiaren sektoreak etengabeko berrikuntza eskatzen du osasun arloko erronkei aurre egiteko. Duela gutxiko COVID-19 pandemiak, biztanleriaren zahartzeak, mendekotasun-tasen gorakadak edota asistentzia sanitario pertsonalizatua sustatzeko beharrak agerian uzten dute monitorizazio eta diagnosirako gailu mediko gero eta sofistikatuagoak, fidagarriagoak eta konektatuak garatzeko beharra. Beti ere modu azkar eta eraginkor batean.

Konektatutako sistema txertatuak funtsezko teknologia bihurtu dira kostu baxuan eta modu azkarrean diagnosi gailu berritzaileak garatzeko. Sistema txertatuak bai monitorizazio eta baita diagnosi sistemetan aurkitu ditzakegu. Horren adibide dira glukosa-monitoreak, taupada-markagailuak, monitorizazio eramankorrerako gailuak, eta abar.

Sektorean dauden aukerez jabetuta, gero eta gehiago dira gailu medikuen negozioan sartzen diren "biotech start-ups" izenez ezagutzen diren enpresak. Ideia eta soluzio tekniko onak izan arren, horietako askok porrot egiten dute, osasun-sektorea eta bete beharreko baldintza erregulatzaileak ezagutzen ez dituztelako.

Gailu mediko txertatuen diseinu eta garapena bi erregelamenduren bidez araututa dago Europan: 2017/745, Medical Device Regulation (MDR) eta 2017/746, In Vitro Medical Device Regulation (IVDR). Era berean, badira beste estandar batzuk, honako alderdi hauek definitzen dituztenak: gailuaren segurtasun funtzionala (IEC 60601), softwarearen bizi-zikloa (IEC 62304), kalitatearen kudeaketa (ISO 13485), arriskuen kudeaketa (ISO 14971), gailuen erabilgarritasuna (IEC 62366) eta zibersegurtasuna (IEC 81001-5-1), besteak beste.

Baldintza arautzaile asko daudenez, beharrezkoa da prozedura-metodologia bat izatea garapen horiek gauzatzeko. Produktu berriak diseinatzeko dauden metodologiak aztertu ondoren, ez da bat bera ere identifikatzen gailu medikoak garatu nahi dituzten start-up motako enpresen beharretara erabat egokitzen denik.

Horregatik, ekarpen nagusi gisa, tesi honek gailu mediko txertatuak diseinatzeko eta garatzeko jarraitu beharreko urratsak zehazten dituen metodologia aurkezten du. Start-up-etarako metodologia denez, arrisku teknologiko handiko etapetan egin beharreko inbertsio ekonomikoa minimizatzea bilatzen da. Gainera, bezeroak edo produktu kudeatzaileak gailua garapenaren etapa goiztiarretan ebaluatzea sustatzen da.

Proposatutako metodologia, medikuntza-sektoreko start-up-etarako gida izan nahi duena, hainbat gailu mediko garatzeko prozesuetan aplikatuz balidatu da. Horietako askok dagoeneko lortu dute CE marka, eta horrek agerian uzten du metodologia egokia dela. Halaber, metodologiaren prozedura kritikoenak Société Générale de Surveillance (SGS) erakunde ziurtatzaileak balioztatu ditu, Teknikerren, ikerketa eta garapeneko zentroan, egindako ISO 13485 ziurtapen-prozesuaren barruan. Azkenik, proposatutako metodologiaren oinarrizko printzipioak balidatzen dituzten hainbat azterlan berrikusi eta aurkeztu dira.

Resumen

El sector de la biotecnología demanda la innovación constante para hacer frente a los retos del sector sanitario. Hechos como la reciente pandemia COVID-19, el envejecimiento de la población, el aumento de las tasas de dependencia o la necesidad de promover la asistencia sanitaria personalizada tanto en entorno hospitalario como domiciliario, ponen de manifiesto la necesidad de desarrollar dispositivos médicos de monitorización y diagnostico cada vez más sofisticados, fiables y conectados de forma rápida y eficaz.

Los sistemas embebidos conectados se han convertido en tecnología clave para el diseño de soluciones de diagnóstico innovadoras de bajo coste y de forma rápida. Estos se encuentran en sistemas embebidos de monitorización y diagnóstico como monitores de glucosa, marcapasos, wearables para monitorización, etc.

Conscientes de la oportunidad que existe en el sector, cada vez son más las denominadas "biotech start-ups" las que se embarcan en el negocio de los dispositivos médicos. Pese a tener grandes ideas y soluciones técnicas, muchas de estas start-ups terminan fracasando por desconocimiento del sector sanitario y de los requisitos regulatorios que se deben cumplir.

El diseño y desarrollo de dispositivos médicos embebidos está regulado en Europa por dos reglamentos: 2017/745, Medical Device Regulation (MDR) y 2017/746, In Vitro Medical Device Regulation (IVDR). Además, existen otros estándares que definen aspectos como la seguridad funcional (IEC 60601), ciclo de vida del software (IEC 62304), gestión de la calidad (ISO 13485), gestión de riesgos (ISO 14971), usabilidad (IEC 62366) y ciberseguridad (IEC 81001-5-1) entre otros.

La gran cantidad de requisitos regulatorios hace que sea necesario disponer de una metodología procedimental para ejecutar dichos desarrollos. Tras realizar un análisis de las metodologías de diseño de producto existentes, no se identifica ninguna que se adapte por completo a las necesidades de empresas tipo start-ups que deseen desarrollar dispositivos médicos.

Por ello, como aportación principal, esta tesis presenta una metodología para el diseño y desarrollo de dispositivos médicos embebidos. Al tratarse de una metodología para start-ups, se busca minimizar la inversión económica a realizar en etapas con riesgo tecnológico elevado. Además, se fomenta la evaluación por parte del cliente o gestor del producto en etapas tempranas del desarrollo.

La metodología propuesta, que pretende ser una guía para las empresas de nueva creación del sector médico, se valida mediante su aplicación en los procesos de desarrollo de diferentes dispositivos médicos. Muchos de ellos ya han obtenido el marcado CE, demostrando así la eficacia de la metodología. Asimismo, los procedimientos más críticos de la metodología son validados por la entidad certificadora Société Générale de Surveillance (SGS) dentro del proceso de certificación ISO 13485 llevado a cabo en Tekniker, centro de investigación y desarrollo. Por último, se revisan y presentan diferentes estudios que avalan los principales conceptos de la metodología propuesta.

Content Index

Abstract **5**

Laburpena **7**

Resumen **9**

Content Index **11**

Figure Index **19**

Table Index **23**

1. Introduction **25**

1.1 Medical Devices and Regulatory Landscape 26

1.1.1 Market pathway 29

1.2 Innovation in Health Technology 30

1.3 Embedded Systems in Healthcare 34

1.4 New Medical Product Design and Development 38

1.4.1 Start-up design culture 40

1.5 Objectives and Structure of this Thesis 43

2. Embedded Medical Devices **49**

2.1 Typical Architecture 50

2.2 Embedded Processors 50

2.2.1 Microcontroller (MCU) 51

2.2.2 Microprocessors (MPU) 53

2.2.3 Digital Signal Processor (DSP) 55

2.2.4 Field Programmable Gate Array (FPGA) 56

2.2.5 System on Chip (SoC) 56

2.3 Embedded Operating System (OS) 57

2.3.1 OS for Microcontroller 58

2.3.2 OS for microprocessors 58

2.4 Memory 61
2.4.1 Volatile Memory 62
2.4.2 Non-volatile Memory 62
2.5 Battery Powered Devices 63
2.6 User Interface 66
2.6.1 Screen 67
2.6.2 Touch panel 68
2.6.3 Haptic systems 70
2.7 Communications 71
2.7.1 Health Level Seven (HL7) 72
2.7.2 ISO/IEEE 11073 73
2.7.3 Point-of-Care Connectivity (POCT1-A2) 75
2.7.4 Communication interfaces 77
2.8 Measurement or Monitorization Unit 80
2.9 Technical Requirements for Medical Device Design and Development 81

3. European Medical Device Regulation 83

3.1 Embedded Medical Device Regulatory Requirements 84
3.2 Medical Device Regulation (MDR) 86
3.2.1 Structure of the MDR 87
3.2.2 Medical device classification according to MDR 89
3.2.3 Pre-commercialization requirements 90
3.2.4 Post-commercialization requirements 92
3.2.5 Traceability and registration requirements 94
3.2.6 MDR Requirements for Medical Device Design and Development 95
3.3 In-Vitro Medical Device Regulation (IVDR) 95
3.3.1 Structure of the IVDR 96
3.3.2 In-Vitro Medical device classification 97
3.3.3 Pre-commercialization requirements 97
3.3.4 Post-commercialization requirements 99

3.3.5 Traceability and registration requirements 99
3.3.6 IVDR Requirements for In Vitro Medical Device Design and Development 99
3.4 IEC 60601 - Basic Safety and Essential Performance 99
3.4.1 Structure of the IEC 60601 series standard 102
3.4.2 Scope of IEC 60601-1 103
3.4.3 Device classification 104
3.4.4 General requirements of IEC 60601-1 105
3.4.5 Safety Requirements for Medical Device Design and Development 107
3.5 IEC 62304 - Medical Device Software 108
3.5.1 Structure of the IEC 62304 standard 108
3.5.2 Scope of IEC 62304 109
3.5.3 Software safety classification 109
3.5.4 Requirements, processes and tasks 109
3.5.5 Software Requirements for Medical Device Development 117
3.6 ISO 14971 - Medical Device Risk Management 119
3.6.1 Structure of ISO 14971 119
3.6.2 Scope of ISO 14971 120
3.6.3 General requirements and risk management processes of ISO 14971 120
3.6.4 Risk Management Requirements for Medical Device Design and Development 123
3.7 IEC 62366 - Medical Device Usability 124
3.7.1 Structure of the IEC 62366-1 standard 125
3.7.2 Scope of IEC 62366-1 126
3.7.3 Categories of user actions 126
3.7.4 General requirements and usability engineering process of IEC 62366-1 127
3.7.5 Usability Requirements for Medical Device Development 130

3.8 ISO 13485 - Medical Device Quality Management Systems 131
3.8.1 Structure of the ISO 13485 132
3.8.2 Scope of ISO 13485 132
3.8.3 Requirements and quality management system processes 133
3.8.4 Quality Management Requirements for Medical Device Design and Development 141
3.9 IEC 81001-5-1 - Medical Device Cybersecurity 142
3.9.1 Structure of the IEC 81001-5-1 143
3.9.2 Scope of IEC 81001-5-1 144
3.9.3 Requirements, processes and tasks 145
3.9.4 Cybersecurity Requirements for Medical Device Design and Development 150
3.10 Batteries Regulation 152
4. Product Design Methodologies 155
4.1 Heavyweight Methodologies 156
4.1.1 Waterfall methodology 156
4.1.2 V-Model methodology 159
4.1.3 Incremental model 162
4.1.4 Iterative model 163
4.1.5 Spiral model 165
4.1.6 Prototype model 167
4.2 Agile Methodologies 169
4.2.1 Key Agile concepts 171
4.2.2 Scrum framework 172
4.2.3 Kanban framework 175
4.2.4 Crystal frameworks 178
4.3 Lean Startup Methodology 181
4.4 Design Thinking Methodology 184
4.5 State-Gate® Methodology 186
4.6 Hybrid Methodologies 189

4.6.1 Scrumban methodology 189
4.6.2 Agile-Stage-Gate 190
4.6.3 Design-Thinking-Lean Startup-Agile 191

5. Methodology Proposal 193

5.1 Requirements 194
5.1.1 Technical requirements 194
5.1.2 Regulatory requirements 195
5.1.3 Methodology requirements 199
5.2 Methodology Proposal 201
5.2.1 Methodology audience 201
5.2.2 Methodology stages 203
5.3 Phase 1 - Development Feasibility 205
5.3.1 Roles 206
5.3.2 Deliverable: Development Feasibility Report 208
5.3.3 Development Feasibility Checkpoint 209
5.4 Phase 2 – Incremental and Iterative Prototyping 210
5.4.1 Overall overview of the proposed model and its deliverables 214
5.4.2 Project Development Plan [PROJ_DEVPLAN] 222
5.4.3 User Requirement Specification (URS) [PROJ_URS] 227
5.4.4 Transference Verification Plan [PROJ_TRANSFER_VERIFPLAN] 228
5.4.5 Risk Analysis [PROJ_RSK] 231
5.4.6 Software [PROJ_SRS] and Hardware [PROJ_HRS] Requirement Specification 236
5.4.7 Software [PROJ_SW_FUNCT_VERIFPLAN] and Hardware [PROJ_HW_FUNCT_VERIFPLAN] Functional Verification Plan 238
5.4.8 Software [PROJ_CHKPOINT_SRS] and Hardware [PROJ_CHKPOINT_HRS] Specification Checkpoint 240

5.4.9 Software [PROJ_SDS] and Hardware [PROJ_HDS] Design Specification 241
5.4.10 Software [PROJ_SW_DEVT_VERIFPLAN] and Hardware [PROJ_HW_DEVT_VERIFPLAN] Development Verification Plan 243
5.4.11 Software [PROJ_CHKPOINT_SDS] and Hardware [PROJ_CHKPOINT_HDS] Design Checkpoint 250
5.4.12 Software [PROJ_SW_SOURCE] and Hardware [PROJ_HW_SOURCE] Implementation..... 251
5.4.13 Software [PROJ_SW_DEV_VERIFREP] and Hardware [PROJ_HW_DEV_VERIFREP] Development Verification Report 252
5.4.14 Software [PROJ_SW_FUNCT_VERIFREP] and Hardware [PROJ_HW_FUNCT_VERIFREP] Functional Verification Report 253
5.4.15 Transference Verification Report [PROJ_TRANSFER_VERIFREP] 254
5.4.16 Configuration Management [PROJ_COMP_CNF] and Baseline [PROJ_BASELINE]..... 254
5.4.17 Change Control [PROJ_CHANGE_CTRL] 257
5.4.18 Incremental and Iterative Prototyping Checkpoint 258
5.5 Phase 3 – Medical Product Consolidation 259
5.5.1 Overall overview of the proposed model and its deliverables..... 260
5.5.2 Project Development Plan [PROJ_DEVPLAN] 266
5.5.3 Validation Plan [PROJ_VALID_TESTPLAN] and Report [PROJ_VALID_TESTREP] 267
5.5.4 Medical Product Consolidation Checkpoint..... 271
6. Methodology Validation.....273
6.1 Successful Methodology Use Cases..... 274
6.1.1 Use Case 1: Medical Device - Wearable Device Sensor 274
6.1.2 Use Case 2: Medical Device for Professional Use 292
6.1.3 Use Case 3: In-Vitro Medical Device for Professional Use 292

6.1.4 Use Case 4: In-Vitro Medical Device for Home and Professional Use 293
6.1.5 Use Case 5: In-Vitro Medical Device for Professional Use 294
6.1.6 Other Use Cases 294
6.2 Certification of the Methodology 295
6.3 Studies Supporting the Proposed Methodology 295
6.3.1 Feasibility phase 295
6.3.2 Agile methodology for iterative and incremental prototyping 296
6.3.3 V-model methodology for medical product consolidation 298
7. Conclusions and Future Works 301
7.1 Conclusions 302
7.2 Principal Contributions 304
7.3 Thesis Articles 307
7.3.1 Embedded Sensor Systems in Medical Devices: Requisites and Challenges Ahead 307
7.3.2 Monitoring of vital signs in the home environment: A review of current technologies and solutions 308
7.3.3 Medical Devices with Embedded Sensor Systems: Design and Development Methodology for Start-Ups 308
7.3.4 Cutting-edge innovation on Artificial Intelligence for integrated care on neuromusculoskeletal disorders 309
7.4 Future Works 311
8. References 313

Figure Index

Figure 1.1-1: Regulatory authorities around the world. 27
Figure 1.2-1: Healthcare Technology Innovation, a timeline.....31
Figure 1.4-1: New medical product development phases. 39
Figure 1.5-1: Thesis outline. 47
Figure 2.1-1: Typical embedded medical device block diagram. 50
Figure 2.2-1: (a) Von Neumann and (b) Harvard architecture. 54
Figure 2.5-1: Battery Management System block diagram. 65
Figure 2.6-1: Display viewing angle. 67
Figure 2.6-2: Resistive touch panel structure.....69
Figure 2.6-3: Capacitive touch panel structure. 70
Figure 2.6-4: Haptic system block diagram. 71
Figure 3.1-1: Embedded medical device modules and their regulations. 85
Figure 3.5-1: Software development processes and activities. 110
Figure 3.5-2: Software maintenance processes and activities.....115
Figure 3.6-1: Main components of the risk management process. 121
Figure 3.7-1: Example of a usability engineering project. 130
Figure 3.8-1: ISO 13485 quality management system's process diagram.....133
Figure 3.8-2: Design and development process phases 137
Figure 3.9-1: Health Software definition. 144
Figure 4.1-1: Waterfall model. 158
Figure 4.1-2: V-model methodology for product design and development.....160
Figure 4.1-3: VDI 2206 V-model design methodology [257]. 161
Figure 4.1-4: Incremental model and its outputs. 162
Figure 4.1-5: Iterative model and its four outputs identified as (1)-(4). 164

Figure 4.1-6: Spiral model. 166
Figure 4.1-7: Prototype model. 167
Figure 4.2-1: Example of a Scrum board. 174
Figure 4.2-2: Scrum workflow. 175
Figure 4.2-3: Crystal workflow. 180
Figure 4.3-1: Lean Startup cycle. 183
Figure 4.4-1: SAP's approach to Design Thinking [300]. 186
Figure 4.5-1: State-Gate methodology workflow. 188
Figure 4.6-1: Scrumban workflow. 190
Figure 4.6-2: Agile-Stage-Gate hybrid model [315]. 191
Figure 4.6-3: Generic Design Thinking-Lean Startup-Agile process. 192
Figure 5.2-1: Scope of the proposed methodology. 202
Figure 5.2-2: The stages of the proposed methodology. 204
Figure 5.3-1: Development Feasibility phase: characteristics, inputs and outputs. 205
Figure 5.4-1: Incremental and Iterative prototyping phase: key characteristics, inputs and outputs. 210
Figure 5.4-2: Proposed Agile methodology framework. 211
Figure 5.4-3: Proposed Agile methodology dashboard. 214
Figure 5.4-4: Proposed V-model for phase 2. 215
Figure 5.4-5: Simplified model for phase 2, V-model with iterations. 216
Figure 5.4-6: Deliverables generated during phase 2. 222
Figure 5.4-7: Diagram of processes to be re-executed after design changes. 258
Figure 5.5-1: Medical Product Consolidation phase: key characteristics, inputs and outputs. 259
Figure 5.5-2: proposed V-model for phase 3. 261
Figure 5.5-3: Deliverables associated with phase 3. 263
Figure 6.1-1: First prototype to validate feasibility. 285
Figure 6.1-2: Assembly to validate the feasibility of the PPG measurement. 286

Figure 6.1-3: Measurements made with PPG and ECG sensor. ..287
Figure 6.1-4: Front side of the device. 288
Figure 6.1-5: Rear side of the device.....289
Figure 6.1-6: Device sizes.....289
Figure 6.1-7: Device placed on the wrist.....290
Figure 6.1-8: Wristband on the charging station. 291

Table Index

Table 1.5-1: Scientific, Technological, and Industrial contributions of this thesis. 48

Table 2.2-1: ARM Cortex M family comparison [120]. 52

Table 2.2-2: ARM Cortex-A family comparison [122]. 55

Table 2.3-1: Medical OS comparison. 60

Table 2.7-1: Bluetooth Classic and Bluetooth Low Energy comparison. 79

Table 3.1-1: Standards and Amendments reviewed during this thesis. 86

Table 3.2-1: Medical Device classification according to MDR. 90

Table 3.3-1: In-vitro medical device classification according to IVDR. 97

Table 3.4-1: IEC 60601 standard series 100

Table 3.4-2: Current IEC 60601-1 standard editions and amendment. 102

Table 3.5-1: Software development process requirements. 112

Table 5.1-1: Technical Requirements 194

Table 5.1-2: Regulatory requirements. 195

Table 5.1-3: Methodology requirements. 199

Table 5.3-1: Roles and description of activities implemented by the participants of the development feasibility phase. 207

Table 5.3-2: Sections and description of the information included in the development feasibility report deliverable. 208

Table 5.4-1: Master document proposal including all project deliverables of phase 2. 218

Table 5.4-2: Principal responsibilities of each team member 223

Table 5.4-3: Considerations for documentation review and approval. 223

Table 5.4-4: Items from the project coordination section of the Project Development Plan deliverable. 224

Table 5.4-5: URS definition. 228
Table 5.4-6: Transference test definition. 230
Table 5.4-7: URS and Transference Test Traceability Matrix definition. 231
Table 5.4-8: Proposed assessment of severity, occurrence and detectability. 233
Table 5.4-9: Proposal for RPN assessment. 234
Table 5.4-10: Risk matrix (before mitigation actions) 236
Table 5.4-11: Risk matrix (after mitigation actions). 236
Table 5.4-12: SRS and HRS definition. 238
Table 5.4-13: Functional test definition. 239
Table 5.4-14: SRS/HRS and Functional Test Traceability Matrix definition. 240
Table 5.4-15: SDS and HDS definition. 243
Table 5.4-16: Development test plan definition. 245
Table 5.4-17: SDS/HDS and Development Test Traceability Matrix definition. 245
Table 5.4-18: Unit Testing Harnesses. 246
Table 5.4-19: Development test report definition. 253
Table 5.4-20: Functional test report definition. 253
Table 5.4-21: Transference test report definition 254
Table 5.4-22: Baseline definition. 257
Table 5.5-1: Master document proposal, including all project deliverables. 263
Table 5.5-2: Validation test definition. 269
Table 5.5-3: URS and Validation Test Traceability Matrix definition. 270
Table 5.5-4: Validation test report definition. 271
Table 6.1-1: Wearable's technical specification. 288

1. Introduction

This chapter aims to set the basis and scope of this thesis. Its objective is to contribute to the development of embedded medical devices. To this end, firstly, the basic concepts related to the development of medical devices are defined, specifically, what a medical device is and the regulatory framework to which its development is subjected before it can be launched to the market. Next, it highlights the need to innovate in health technology, showing how the socio-economic and technological environment of the last few years has encouraged leading sector companies to innovate more and more. It also explains how the number of start-ups in the industry has been increasing yearly. Likewise, the importance of embedded systems within this sector is evidenced, presenting the advantages and disadvantages of these systems. Furthermore, the phases of design and development of a medical product are defined, pointing out the key aspects and the problems faced by the different types of companies. Particular emphasis is placed on the complexity of developing medical devices for start-up companies and why many fail. Finally, both the objectives and the structure of this thesis are detailed.

1.1 Medical Devices and Regulatory Landscape

The World Health Organization (WHO) defines the **Health Technology** concept as the application of organized knowledge and skills in devices, medicines, vaccines, procedures and systems developed to solve a health problem and improve quality of life [1].

However, more and more organisations and publications are differentiating between the concepts HealthTech and MedTech since there is no clear consensus on the definition of each term. As a result, Jorge Juan Fernandez, Innovation Director at EIT Health, published an article in May 2021, collecting different meanings for HealthTech [2]. This article presents that HealthTech mainly refers to telemedicine and remote patient monitoring services such as web platforms or mobile applications used for patient care. Similarly, biometrics and some wearable devices are also considered under this category. In contrast, Medtech focuses more on technologies used to treat a condition or diagnose a disease. This concept includes, for example, medical devices and equipment for diagnosing diseases and illnesses.

Due to the lack of agreement on the definition, this document will assume the definition made by WHO as appropriate, understanding HealthTech as everything related to the care and improvement of patients' quality of life.

It can be considered undisputed that HealthTeach plays a relevant role during all stages of our lives as it includes many much-needed products and services. Latex gloves, thermometers, blood pressure monitors or mobile applications to manage primary health care appointments are considered part of HealthTech. Medical devices have become a key element in any healthcare system. These devices help healthcare professionals properly monitor, diagnose and treat their patients.

The medical device industry is regulated by different national notifications or regulatory bodies. The world's two main regulatory bodies are the European Commission Directorate and US Food & Drugs Administration (FDA). Figure 1.1-1 shows the most relevant regulatory authorities around the world.

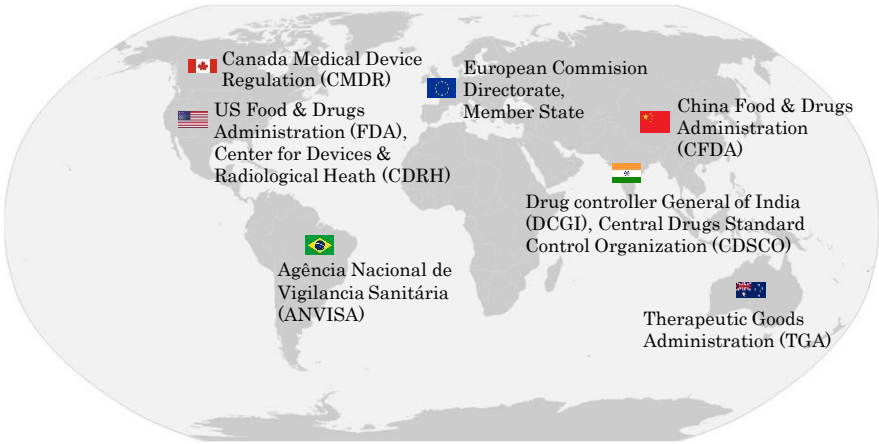


Figure 1.1-1: Regulatory authorities around the world.

In the European Union, medical devices are regulated by harmonised health regulations. Any manufacturer wanting to put a device on the European market must pass through a notified body to assess and approve its device. If it is considered approved, a certificate of conformity with the CE mark is emitted, which allows it to be sold in all the countries of the European Union.

According to the European Medical Device Regulation (MDR), the following are considered medical devices "Any instrument, apparatus, appliance, software, implant, reagent, material or other article intended by the manufacturer to be used, alone or in combination, for human beings for one or more of the **following specific medical purposes: Diagnosis, prevention, monitoring, prediction, prognosis, treatment or alleviation of disease, diagnosis, monitoring, treatment, alleviation of, or compensation for, an injury or disability, investigation, replacement or modification of the anatomy or of a physiological or pathological process or state**, providing information by means of in vitro examination of specimens derived from the human body, including organ, blood and tissue donations and which does not achieve its principal intended action by pharmacological, immunological or metabolic means, in or on the human body, but which may be assisted in its function by such means [...]" [3].

In contrast, the U.S. Food and Drug Administration (FDA) considers a medical device "An instrument, apparatus, implement, machine,

contrivance, implant, in vitro reagent, or other similar or related article, including a component part, or accessory which is:

(A) recognized in the official National Formulary, or the United States Pharmacopoeia, or any supplement to them,

(B) intended for use in the diagnosis of disease or other conditions, or in the cure, mitigation, treatment, or prevention of disease, in man or other animals, or

(C) intended to affect the structure or any function of the body of man or other animals, and which does not achieve its primary intended purposes through chemical action within or on the body of man or other animals and which does not achieve its primary intended purposes through chemical action within or on the body of man or other animals and which is not dependent upon being metabolized for the achievement of its primary intended purposes" [4].

In both cases, it is evident that in order to discern whether a product is a medical device, it is necessary to define its **intended use** [5]. For example, smartwatches, which have become so common recently, depending on their intended use, can be considered medical devices. A smartwatch that measures the pulse or performs an electrocardiogram is considered a medical device only if the readings are used to diagnose a disease or establish treatment [6]. On the other hand, with the same technology, this smartwatch is not considered a medical device if it is only used for informative purposes; this is the case when a user is interested in measuring the pulse rate after a workout. This non-medical device category is appropriate for devices such as the Apple Watch [7].

Medical devices can be classified into different categories based on their inherent risks. According to the MDR, the risk is categorised from lower to higher in Europe as class I, IIa, IIb and III. Furthermore, depending on the risk presented by each device, different requirements must be met to obtain the CE marking.

The FDA defines three categories of medical devices: Class I, II and III. Class I includes devices with a low-risk level, Class II with a medium-risk level and Class III high-risk level. In addition, FDA defines a new concept in device classification, "predicate device".

This agency allows the assignment of the risk level by comparison. In this way, device manufacturers can claim that the device submitted is equivalent to a previously marketed one simplifying the approval process. The predicate device concept benefits manufacturers who develop new generations of their devices.

The design and development of these systems require considering stringent regulations from the conception phase of the idea. In [8], a study is presented detailing the challenges of developing medical devices according to FDA regulations. Regardless of country or region, the design and development of medical devices are regulated by several standards that guarantee the quality of the devices and minimise the exposition risk to healthcare professionals and patients [9], [10].

The regulations to be applied and their interpretation depend on the medical device's intended use and technologies. For example, there are purely software devices [11], including software and electronics, or only mechanical devices. Common and specific regulations exist for all these cases [12].

1.1.1 Market pathway

In all industries, manufacturers of devices or equipment aim to get their products to market as early as possible. In the medical sector, bringing a product to market is lengthy. Medical devices, as they act directly or indirectly on the state of health of humans, require special attention. Risk and quality management are essential to ensure the quality of the product delivered to the market.

In Europe and the U.S., the regulatory path to the market is determined by the device's classification. Depending on the type of device, this will be a more or less complex process.

In Europe, to place a device on the market, the manufacturer must declare the conformity of the product with the general safety and performance requirements established for its type of device. Safety and performance must be demonstrated through conformity assessment procedures specified according to the risks involved in their product. At least, the design must comply with IEC 60601-1 (Medical electrical equipment, general requirements for basic safety and essential performance) and the software design with IEC 62304

(Medical devices, software life cycle processes). In some cases, the demonstration of compliance must include a clinical evaluation.

When the device is categorised as low risk, class I, it is enough to generate a self-certification of conformity to obtain the CE marking. However, suppose the risk is moderate and high. In that case, a notified body must perform the device's conformity assessment and the manufacturer's quality system evaluation.

In the U.S., 510(k) or Premarket Approval (PMA) procedures are required depending on the device's class. The 510(k) applies when it is possible to demonstrate the equivalence of the medical device with another device already on the market. In contrast, the PMA process applies when a scientific and regulatory review is necessary to assess the risk and effectiveness of the device to be commercialised.

The healthcare sector requires more medical devices to help professionals diagnose and treat patients. Developing these devices is not an easy task as these devices are regulated under strict regulations.

1.2 Innovation in Health Technology

The medical industry has constantly been evolving over the last few years. On the one hand, new healthcare challenges are emerging, such as COVID-19 or the problem of an elderly society [13]. On the other hand, the rapid evolution of technology is making it possible to improve current medical devices and solutions.

COVID-19, a disease caused by the new coronavirus known as SARS-CoV-2, emerged on 31 December 2019 in Wuhan (China) [14]. Despite becoming a global pandemic, COVID-19 has introduced technological innovation in the healthcare sector. Telemedicine had come to remain; before COVID-19, it was usual to go to the doctor face-to-face [15]. However, due to the collapse of health systems, mobile applications and information systems for patient care have been developed worldwide. Likewise, disinfecting robots [16], devices for monitoring temperature in public spaces [17] or low-cost oximeters [18] for home use are clear examples of the technological evolution this pandemic has brought about.

The problem of an ageing population is another challenge to be faced. The increase in life expectancy and the considerable decrease in the birth rate make it essential to take measures to help manage and optimise patient care. The WHO estimates that between 2015 and 2050, the world's population over the age of 60 will increase from 900 million to 2 billion [19]. In this context, technological developments oriented towards patient monitoring in home and hospital environments are particularly relevant.

Not only the emergence of new challenges in the sector has brought new technological innovations. The development of technology in aspects such as the Internet of Things (IoT) and Artificial Intelligence (AI) also makes it possible to develop many innovative Medical Technologies [20]. As a result, new solutions and devices have appeared in the healthcare sector, allowing (i) to prevent diseases or damages [21], [22], (ii) to diagnose diseases or special conditions [23], (iii) to monitor the patient's condition [24], [25], (iv) to help treat and overcome diseases [26], [27] and (v) to care for and facilitate the process of patient recovery [28], [29]. Figure 1.2-1 presents some of the most relevant innovations in the 1980s and 1990s.

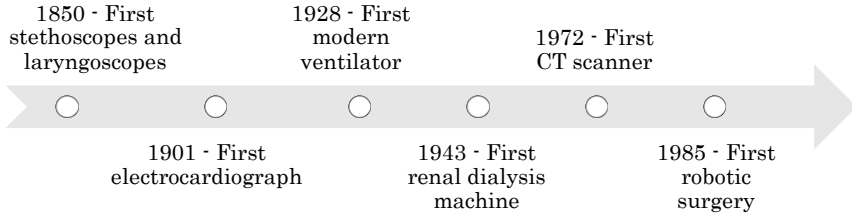


Figure 1.2-1: Healthcare Technology Innovation, a timeline.

Currently, some of the advances and developments that are revolutionising the medical sector include:

- Data management optimisation:

An increasing number of mobile or portable devices are being used to monitor patient parameters such as temperature, oxygen or pulse rate [30]. Storing, organising and analysing all this data is not an easy task. However, thanks to Big Data and AI, it is possible to manage vast volumes of data efficiently [31].

The wearable system proposed in [32] can diagnose diabetes using machine learning and big data. In [33], a Big Data system is developed to support the rehabilitation of strokes and lung diseases. The heterogeneity of data capture systems leads to the development of architectures to support such different solutions. In [34], a semantic Big Data architecture to address the heterogeneity of data between several wearable platforms is presented.

- Artificial vision systems:

These systems are excellent enablers for developing new approaches in the healthcare field. Complex systems, such as endoscopes, radiology, ophthalmology, surgery, etc., use this technology.

Moreover, embedded vision, based on integrating directly adapted camera modules into medical devices, enables intelligent image processing in various portable applications. One is eye-tracking systems, which can be used for diagnostics or patient care. In [35], a vision algorithm is presented to detect eye movement to identify ocular pathologies, such as strabismus. In [36], an algorithm is proposed that, together with the Irisbond eye-tracker [37], can assess mathematics in children with cerebral palsy. Eye-tracking systems can even be helpful for healthcare professionals. Article [38] presents the possible use of eye-detection systems to assist neonatal resuscitation. In [39], a pilot study is shown for the same purpose.

- Early diagnosis of diseases:

In a few years, Artificial Intelligence (AI) will make it possible to diagnose diseases such as lung cancer [40]. The analysis of thousands of digital scans will identify early stages of cancer that would not have been possible with traditional technology [41].

In [42], different AI algorithms are reviewed to diagnose and treat prostate cancer. Similar analyses are also supported by [43] for colorectal cancer detection and [44] for breast cancer detection.

- Patient monitoring at home and in the hospital:

Vital signs monitoring allows the patient's progress to be evaluated and ensures early detection of undesirable effects. Advances in embedded electronics with integrated sensorisation allow reliable

temperature, oxygen, pulse or blood pressure measurements using comfortable, self-powered devices.

More and more work is being performed on developing wearable solutions, such as smartwatches, that allow continuous patient monitoring in a non-invasive way [45]. Article [46] presents a system capable of measuring heart rate, SpO2 and respiratory rate. It is a low-cost system, which makes it interesting for deployment in low-resource settings. There are also solutions in the literature that can detect falls [47], [48]. These systems are especially interesting for elderly or fragile patients [49].

In this context, leading medical companies' Research and Development (R&D) strategies indicate the need to evolve and develop current technology. For this purpose, many have alliances with universities and research centres in which they invest a large percentage of their annual revenues. Roche, the world's largest biotech company, with revenues of about 55 billion Switch francs, invests in R&D around 9 billion Swiss francs every year, one of the highest innovation spending figures across all sectors [50]. Medtronic, a leading manufacturer of medical technologies whose portfolio includes infusion pumps, medical devices and advanced electrical instrumentation for surgery, with a revenue of about $30 billion, invests more than $2.5 billion in R&D each year [51]. Other leading companies in the sector, such as Alcon [52], Siemens [53] or Abbott [54], also invest around 10% of their turnover in R&D.

Medtronic launched the first patient procedure with the Hugo RAS robotic-assisted surgery system in 2021. This platform includes AI technology that records and processes images from the operating room [55]. Roche, through its collaboration with Microsoft since 2017, is transforming in vitro diagnostics with solutions based on the Microsoft Azure IoT Platform. This way, Roche achieves intelligent and remote management of its in vitro devices [56]. Based on the success of this collaboration, in December 2021, Roche and Microsoft signed a new agreement to integrate AI and cloud technology into their devices [57].

The evolution of technology and the multitude of programmes promoted by the administrations to improve citizens' healthcare is

generating a massive wave of start-up companies that aim to design, develop, and put new medical devices on the market. Moreover, according to a report published by the Spanish Association of Business Angels Networks (AEBAN), 40% of the start-ups created in Spain during the last few years belong to the medical sector [58]. It also explains that the most attractive sectors for Business Angels are mobility, healthcare and energy.

Among the Spanish start-ups, Koa Health, Inbrain Neuroelectronics and MedLumics stand out as the start-ups with the most funding in 2021, with the three companies totalling more than 60 million euros [59]. Koa Health works on various mental health solutions, from digital wellness to digital therapy, with the tools it offers to improve users' mental well-being anytime and anywhere [60]. Inbrain Neuroelectronics is developing a minimally invasive neural interface that can detect and modify specific biomarkers using AI and Big Data to improve personalised neurological therapies [61]. MedLumics specialises in developing cardiac optical imaging ablation devices for atrial treatment [62].

Therefore, there is an increasing need to develop new medical devices with sophisticated technical solutions that allow the healthcare sector to evolve. The emerging technologies have enabled the creation of start-ups that aim to introduce new devices into the market, with no need to invest enormous resources in their development.

1.3 Embedded Systems in Healthcare

Embedded systems are electronic devices that are specifically designed to perform certain functions. They are usually highly optimised systems and provide high levels of system integration for developing different devices, manufacturing processes and using goods and services.

Embedded systems are usually composed of hardware, firmware and software. The hardware consists of the physical and electronic components required to fulfil the functionality of the embedded system. The main element is the processing unit which controls other integrated circuits such as memories, analogue-digital

converters, power supplies, or battery controllers. The software is a set of instructions or programs programmed in the processing unit to answer specific use cases or functions of the system. Moreover, the firmware is the set of instructions implemented at the processing unit to control the electronic circuitry. Firmware is considered the link between hardware and software.

Embedded systems have been transforming the healthcare industry over the last few years. An increasing number of embedded devices enable continuous monitoring of vital signs, glucose, etc. References [63] and [64] review different embedded solutions for monitoring vital signs. Several solutions based on smartwatches or even sensors integrated into textiles or lenses are presented. These small and connected devices are making it easier to capture and transmit information to healthcare centres. Article [65] presents how smart embedded systems offer secure, low-cost communication interfaces for healthcare services. Once enough data is available; this information can be post-processed using AI diagnostic algorithms to improve the diagnosis results. In [66], there is evidence that AI can improve the diagnosis of rare diseases. Devices such as pacemakers made by embedded systems are a significant breakthrough for patients with heart disease. These devices can monitor heartbeat and react to cardiac malfunctions [67]. They also register all the data so doctors can adjust the patient's therapy more efficiently.

Sensors for healthcare monitoring are usually devoted to measuring vital signs. Currently, four basic parameters are defined as vital signs [68] blood pressure, heart rate, respiratory rate and body temperature.

According to WHO, blood pressure is *"the force exerted by circulating blood against the walls of the body's arteries"* [69]. Among the solutions for blood pressure measurement, the oscillometric systems can analyse the vibration of the arterial wall based on the signal transduction method. Authors in [27] and [28] present some wearable designs using capacitive sensors. Auscultatory systems integrate microphones to interpret sounds during the measurement process. However, due to their measurement principle, these should be used in low-noise environments [70]. Other methods to estimate blood pressure are presented in [71], a photoplethysmograph based on an optical signal

or the one shown in [72], which determines blood pressure without contact using video analysis.

Heart rate, heart beeps per minute, is commonly measured by electrocardiographs that measure the voltage potential generated by the electrical signals that control the expansion and contraction of the heart. In [73], the author presents the design of a portable electrocardiograph. It can also be measured by optical systems that determine the heart rate by emitting a beam of light into the sub-choroidal vessels and measuring the reflected light in a photosensor. Many wearable developments are based on optical systems as their measurement principle does not require electrodes, making them suitable for these systems [74], [75]. In [76], both measurement methods are compared. This study concludes that although the traditional electrocardiograph-based measurement is the most reliable, the optical system can accurately measure heart rate variability. There are also less precise measurement systems, such as those based on video images. In [77], the author introduces a system using facial photographs. The author in [78] develops a system based on an infrared CMOS camera to measure heart rate.

Respiratory rate is the number of breaths per minute [79]. This parameter can be measured by an impedance spirometer that measures the variation in body resistance during breathing [80]. It is common to measure the respiratory rate through acoustic systems placed on the neck [81]. However, one of the most accurate systems is based on capnography. It measures the concentration of carbon dioxide in the patient's airway to determine the respiratory rate [82]. Nevertheless, it is a contact-based system that cannot always be used. There are also non-contact systems, such as the one presented in [83], which uses a Doppler sensor placed on the ceiling of an intensive care unit. In [84], a system capable of measuring the heart rate with imaging systems is presented.

Finally, body temperature can be measured using different methods. On the one hand, it is possible to carry out such measurements with contact thermometers. These include temperature sensors and predictive algorithms for fast measurement and higher sensitivity. The system presented in [85] is based on continuous temperature measurement in the ear. It combines the reading with statistical learning algorithms for higher accuracy. In [86], the author presents

an intelligent pillow that can estimate body temperature using machine-learning algorithms. It is also possible to measure body temperature without contact; several developments are based on infrared measurement [17], [87].

Embedded systems are used not only in monitoring medical devices; in recent years, they have been extended to all health technology categories [88]. They can be found in diagnostic devices such as blood glucose monitors, blood INR monitors, defibrillators and digital thermometers; prognostic devices such as PET, digital X-ray and MRI; patient management solutions such as self-test devices for remote patient monitoring; and telemedicine applications [89].

Embedded systems in the medical sector have become very interesting, providing many essential advantages. These systems are **highly customizable and controllable**, as the software and hardware design are usually tailor-made for each application. Furthermore, the complete design of the hardware, firmware and software allows the developer to control the system at all times. The author in [90] presents design techniques for lightweight, re-configurable medical systems based on embedded technology. Regarding its cost, these are **low-cost** systems with a dedicated design that makes these systems cost-effective. This feature has opened the door to disposable or widely adopted electronic devices like wearable electronics. The author of [91] highlights the use of embedded systems in countries where funds are tight. Finally, due to being highly optimised systems, response times can be minimal, ensuring **real-time execution**. This feature is critical in the medical sector as it minimises the sanitary reaction time or even the time required to dose the treatment. In [92], the importance of real-time systems in insulin pump devices is evidenced.

On the other hand, these systems also have disadvantages or difficulties that must be carefully monitored in the health sector. These systems, especially those with accessible communication ports, can present **vulnerabilities** as they are susceptible to being hacked [93]. In an industrial or consumer PC-based system, installing anti-virus applications or firewalls that act as a barrier against attacks is possible. Embedded systems are not immune to such attacks, and firewalls and anti-virus applications cannot usually be integrated into the processing unit. Therefore, it is

required for the firmware/software developers and the hardware designers the implementation of advanced security mechanisms [94]. As it is mentioned in [95], it is considered good practice to implement secure update mechanisms, secure key storage elements, data encryption, etc.

In addition, many **free hardware and software packages are available** on the web, enabling almost any technician with some knowledge of electronics to implement solutions based on embedded systems. Although this may seem advantageous, it becomes a problem for the medical sector. These free hardware and software are not usually designed to satisfy the requirements of the healthcare sector and therefore do not comply with the required regulations to commercialise these solutions [96]. Usually, start-ups, due to the lack of specific technical knowledge and unfamiliarity with the medical sector requirements, develop their products based on open-source hardware and software platforms. Eventually, when they attempt to market them, they realise that a complete redesign of the designed device is required.

Developing new medical devices based on embedded systems offers good opportunities for the health sector. However, it is necessary to consider the peculiarities of these systems to design and develop a medical device successfully.

1.4 New Medical Product Design and Development

The development of a new medical device is a complex and resource-intensive process. The complexity of medical device development lies in compliance with the associated regulations. A weak identification of standards and requirements or a technically optimal design without considering regulatory aspects can result in an unsuccessful medical device development. Figure 1.4-1 presents the typical phases of a medical product design and development strategy. These stages are detailed in the following [97]:

- Feasibility: This phase identifies the market needs, clinical and regulatory aspects of the project development, economic impact, etc.

- Design and Development: Functional requirements are identified, and the project plan is established. After this definition, the first prototype is developed. It is common to have design iterations during this process.
- Design verification: Development is verified, and a functional prototype is obtained. The components are ensured to fulfil this stage's established requirements and safety standards.
- Certification and Qualification: In this step, it is necessary to guarantee the product's compliance with the requirements established by the accredited certification and standardisation organisms. This phase includes (i) clinical trials to guarantee compliance with the requirements, which depending on the category of the equipment, are mandatory, (ii) electrical safety tests, electromagnetic compatibility tests, etc., and (iii) the achievement of the product's commercialisation acceptance.
- Industrialisation: The product is transferred to production. For successful industrialisation, this step must include quality assurance control mechanisms.
- Post-market surveillance: Once the product is on the market, monitorization of the device is necessary to identify problems, know user satisfaction, and publish software updates to answer to identified vulnerabilities.

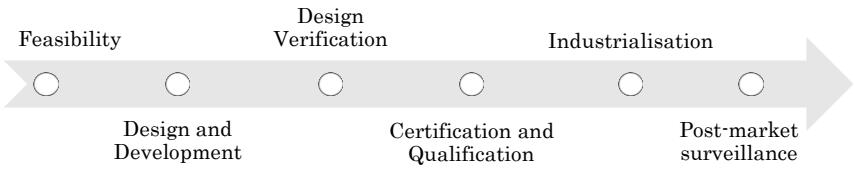


Figure 1.4-1: New medical product development phases.

When developing new medical products, it is mandatory to have a quality management system [98]. European and U.S. regulations require a comprehensive quality management system covering design, development, industrialisation and post-market phases [99].

ISO 13485 defines a quality system that is recognised worldwide. Its compliance proves that the organisation has established the appropriate procedures to design, manufacture and maintain a medical device [100]. To demonstrate compliance with ISO 13485, medical device manufacturers or design centres can obtain the

corresponding certification. The certification of a company in ISO 13485 is not mandatory to design and develop a medical device. However, it eases the product certification process. Implementing ISO 13485 in an organisation is not usually straightforward as it involves establishing procedures for planning, requirements definition, design control, change management, verification and validation, documentation and transfer.

1.4.1 Start-up design culture

Start-ups are essential in the medical device sector as many of them are responsible for the evolution of the healthcare sector with their innovative technological developments. Moreover, large consolidated companies buy many innovative concepts or products created by these companies to continue to evolve and bring new products to the market.

Despite the large number of start-ups that arise every year, according to a study performed by CB Insight, at least 70% of start-ups fail and disappear without reaching the second stage of venture capital investment. Success rates decrease drastically as venture rounds progress, with a final success rate of only 1%. This report also highlights the main reasons for failure: difficulty in obtaining funding (38%), lack of market knowledge (35%), ignorance of competition (20%), flawed business model (19%), regulation challenges (18%), product pricing issues (15%), not having the right team (14%) and failure to manage go-to-market times (10%) [101].

According to CB Insight, the main problem of the development team is the lack of a multidisciplinary team with enough expertise to meet all the technological challenges that arise during the design and development phases. Additionally, the product design does not answer the market's needs, or the development is not competitive. Other failure cases present an uncompetitive cost, costly technology or poorly optimised processes that make these developments go to market with exorbitant prices, making it difficult to place them in the market.

Also, according to [102], since the biotech sector is based on continuous innovation, it is essential for small companies such as start-ups to have strategic partners for innovation. The notable

increase in the number of patents in the biotechnology and medical technology sector evidences the innovative nature of these companies. According to data from the European Patent Office, in 2020, medical technology patents were the leading field, while pharmaceuticals and biotechnology were the fastest-growing sectors [103]. Authors in [102] also underline that strategic alliances between different enterprises should help start-ups in one critical phase of their solutions: the regulatory process. While large, established companies have experience in certification, mass production, commercialization and product maintenance, start-ups are often completely lost when they arrive at these development phases.

Article [104] highlights the difficulty for start-ups to cope with the new regulation that applies to the development of medical devices. In addition, the author of [105] states there is a risk of slowing down innovation because the new MDR requires more costly and high-quality testing. It also requires more technical documentation to comply with the new regulation.

Start-ups, by their nature, tend to focus on quickly generating and bringing to market a first generation of their device. Despite not having consolidated solutions, they often need to accelerate to obtain new funding rounds and thus try to survive. Therefore, this type of company invests many resources during product development phases.

Usually, start-ups must face problems such as managing technical teams that grow very fast or talent loss, issues directly affecting the development phase [106]. These problems, combined with the need to go to market as soon as possible, lead to the carelessness of other steps such as the feasibility and analysis phase, design, verification, certification, industrialisation, and maintenance. In contrast, consolidated companies with structured teams, well-differentiated departments and competencies spend much time analysing and designing the product. The product requirements are clarified and managed during these phases, reducing the development risk. The cost of these stages is usually high, as a consensus between the different departments is often prioritised.

The development of medical devices requires design and development following regulations and quality management systems in which the development phase is not the most critical. Instead, these developments' analysis, design, verification and certification phases are the core. These stages tend to be based on quality management systems, often unmanageable for start-up companies. These systems require many resources in phases where the progress towards obtaining the new product is not evident at first glance. The unawareness of these processes and the difficulty of translating these efforts into progress towards the final goal causes many start-ups to die before they even go to market, even though they have highly innovative technical solutions.

Despite this, start-up companies have characteristics that are difficult to find in consolidated companies. The adaptability to any transformation, the clear identification that innovation is necessary, taking responsibility for the work carried out by each individual in the company, and the hunger for survival are all positive aspects of the start-up culture. These features are rarely identified in consolidated companies and are becoming increasingly important in the current world of uncertainty.

Currently, several methodologies for product development can be aligned with the requirements of ISO 13485. These methodologies are reviewed in section 4. However, all of them are generalists, covering design phases without the definition of documentation to be generated or specific technical aspects of the developments. Not considering documental or technical requirements can result in product non-acceptance by the regulatory authorities.

Medical companies developing medical products for years have enough internal knowledge to deal with all the design phases. On the contrary, start-up companies do not have methodologies that consider all these aspects and, therefore, have a high probability of failing during the design and development phases. Therefore, this thesis aims to cover the lack of specific methodologies for medical product development with embedded software and hardware.

New product development in the medical sector requires the knowledge to identify applicable requirements and regulations

and establish a quality management system to ensure quality during the product's life cycle.
The start-up culture has characteristics that make it challenging to develop new medical devices successfully. Although the concept of innovation and transformation is well understood, the lack of awareness of the relevance of analysis, design and verification phases often causes the death of many start-ups along the market path.

1.5 Objectives and Structure of this Thesis

Considering the growth perspectives of the medical device sector and the evolution of electronics towards more embedded devices, this thesis aims to facilitate the integration of embedded electronics in medical devices. This thesis seeks explicitly to **define and validate a medical device design and development methodology that integrates embedded hardware and software. Specifically, this thesis is intended to be a guide for start-ups seeking to develop devices for healthcare.**

Additionally, the following sub-objectives are identified:

- **Identify the needs of the health sector and the challenges that start-ups face** when developing their products and services.
- **Study and analyse the technical and regulatory requirements** that apply to the design and development of embedded medical devices. In addition, it is intended to define a typical embedded medical device that includes the main components that usually comprise these systems.
- **Study and analyse existing product design and development methodologies**, highlighting their weaknesses and strengths for application in the health sector.
- **Minimise the number of start-ups that fail in the sector** by defining a methodology that guarantees compliance with regulatory requirements, responds to market and customer needs and minimises economic investment in stages with high uncertainty. In addition, the proposed methodology should apply to any typical embedded medical devices.

For this purpose, it is intended to (i) identify the technical and regulatory requirements for medical devices which integrate embedded and connected electronics, (ii) identify current design and development methodologies, (iii) establish a design and development methodology in compliance with international design and development regulatory requirements and (iv) validate the methodology in a real environment.

This work arises from the needs identified by the PhD candidate during her years of experience at the Tekniker research centre. She has managed and developed medical device projects for start-ups and well-established companies in this centre. In addition, she has participated in defining Tekniker's medical product design and development procedures according to ISO 13485, specifically in its implementation and subsequent certification process. All this allows the PhD candidate to have a realistic view of stakeholders' requirements and problems in developing new medical devices.

Chapter 1, Introduction, aims to contextualise and define the purpose of this thesis. To do so, it presents what is considered a medical device and the main difficulties involved during its design and development. Likewise, the health sector is constantly evolving since medical devices must be adapted to the latest medical advances and needs. This innovation is often driven by embedded systems, which have provided a gateway to sophisticated monitoring and diagnostic techniques. However, innovation in medical devices is not only driven by large players in the sector; start-ups also have a very relevant role as an increasing number of such companies are trying to develop medical devices. The truth is that despite having great ideas, many of these companies end up failing. One of the main reasons for this is the lack of understanding of combining technical and regulatory requirements.

Chapter 2, Embedded Medical Devices, defines a typical embedded system for the medical sector. The main blocks and design alternatives are described: embedded processors, embedded operating systems, memories, battery-powered devices, user interfaces and communications. In this section, besides reviewing the most relevant technologies for each component of embedded systems, the specific requirements that must be fulfilled to be used in the medical field are also outlined. This section highlights that

the design of embedded systems for the medical sector must consider specific aspects of the medical field.

Chapter 3, European Medical Device Regulation, reviews the regulations that apply to all embedded medical devices. The MDR and IVDR are identified as the principal European Union regulations for the design and development of embedded medical devices. In addition, other standards are also recommended to be followed. The most extended and applicable to embedded devices include IEC 60601, IEC 62304, ISO 14971, IEC 62366, ISO 13485, IEC 81001-5-1, IEC 62133, IEC 60086-4 and UN 38.3. Once the main standards are reviewed, the regulatory requirements that must be fulfilled during the design and development phases are extracted from each. These requirements must be considered by the methodology defined in this thesis. The reviewed regulatory requirements provide evidence that a methodological approach is needed to assist engineering teams and medical product designers in complying with these requirements efficiently.

Chapter 4, Product Design Methodologies, includes a review of product design methodologies. This review covers traditional approaches, such as the Heavyweight methodologies or more contemporary ones, such as Agile, Lean Startup, Design Thinking or Stage-Gate. Other methods that combine different approaches are also reviewed. The analysis of each methodology highlights its advantages and disadvantages when applying them in the design and development phases of medical devices. Furthermore, the main characteristics that can contribute most to a methodology for start-up companies to design and develop medical devices are extracted from each. The review of methodologies evidence that none of them is fully adapted to the needs of start-ups facing new designs in the medical sector.

Therefore, **Chapter 5, Methodology Proposal**, presents a methodology that meets the needs of start-ups. The proposed method is based on an approach that minimises the required investment and development risk. Likewise, to meet the client's or product manager's expectations, the methodology aims to obtain feedback from them during the early stages of development. To this end, a 3-stage method is defined. The first phase, Development Feasibility, aims to transform ideas into feasible technical solutions.

Therefore, it seeks to carry out a first-solution approach that can clarify the main technical uncertainties within the lowest investment. The second phase, Incremental and Iterative Prototyping, is based on the agile development of all required functionality and a first version of the documentation. This stage is based on developing several prototypes that will ease client feedback. Finally, the third phase, Medical Product Consolidation, aims to strictly comply with the regulatory requirements that apply to the design and development of medical devices. This stage will consolidate the progress made in the previous phases. To this end, special attention is given to the generated documentation and the product verification and validation process.

Chapter 6, Methodology Validation, aims to validate the proposed methodology. First, the use cases in which the proposed methodology has been applied are detailed. Some of the developed systems have already been CE marked as medical devices, evidencing thus the suitability of the method. Moreover, this methodology has been established in Tekniker's Electronics and Communications Unit, where embedded medical devices are designed and developed. These procedures have been evaluated and validated by the certifying entity Société Générale de Surveillance (SGS) within the ISO 13485 certification process of the centre. Finally, state-of-the-art studies supporting the proposed methodology's fundamentals are reviewed and presented.

Finally, **Chapter 7, Conclusions and Future Works** highlights this work's main conclusions and contributions. It also identifies future research lines.

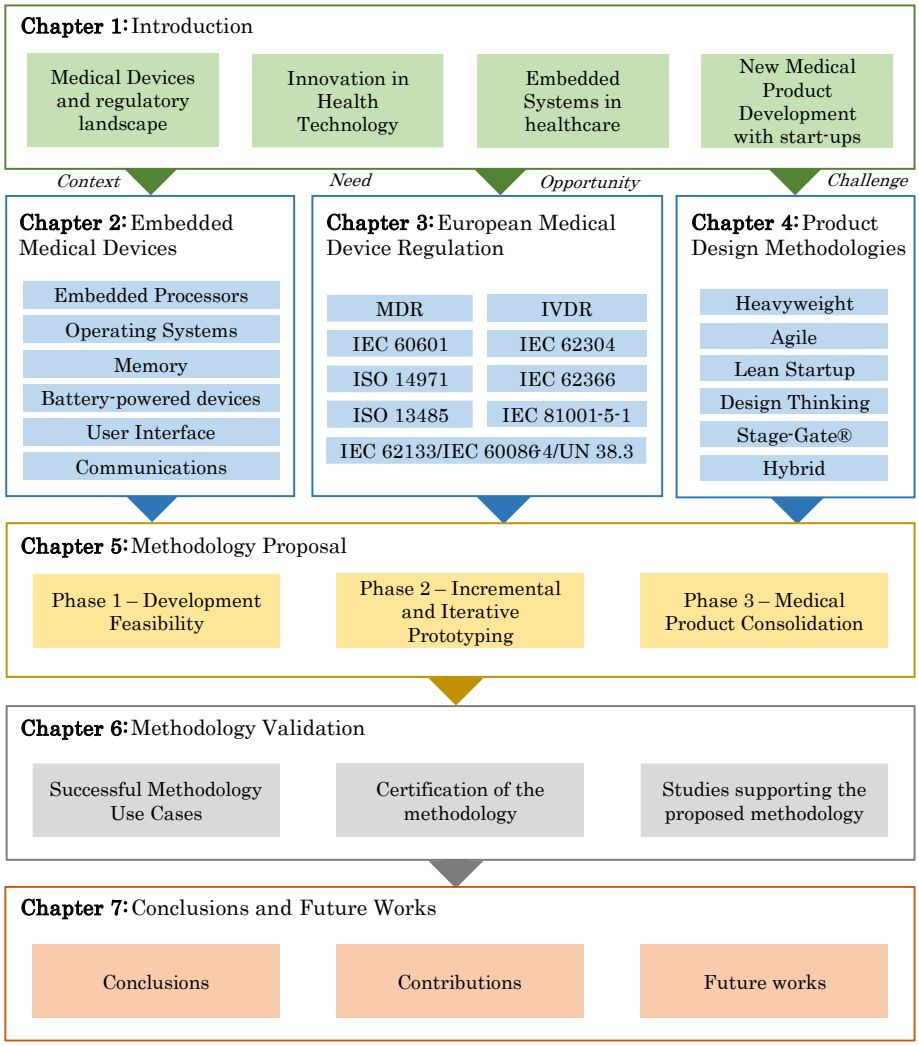


Figure 1.5-1: Thesis outline.

The most relevant scientific, technological, and industrial contributions are listed in Table 1.5-1. Each contribution is related to the respective chapter and publication.

Table 1.5-1: Scientific, Technological, and Industrial contributions of this thesis.

Contribution	Chapter	Publication
Identification of technical and regulatory challenges and requirements applicable to the design and development of embedded medical devices.	2 and 3	[107]
Analysis of existing new product design and development methodologies identifying their incompatibility with medical product design in start-ups.	4, 5 and 6	[64],[108],[109]
Methodology proposal		
Methodology validation through its application in real use cases.		

2. Embedded Medical Devices

Developing embedded medical devices requires understanding the elements of a typical embedded system. Most of these systems share several technological building blocks. Therefore, a detailed analysis of these subsystems is proposed. To this end, this chapter describes the typical architecture and presents the existing technologies for each building block. Different types of embedded processors, the heart of these systems, are analysed and compared. Likewise, the operating systems implemented on embedded processors and used as the core of the medical application are reviewed. Specifically, details of the operating systems available for the different categories of medical devices are given. Data storage is also crucial in any medical device, so the different types of memory are presented. Since these devices can be portable, their power subsystem is described. Also, the various user interface options are discussed. Medical devices are now integrated with hospital communication networks, which is feasible due to different protocols and communication interfaces already accepted and standardised. The main ones are presented in this section. Finally, the measurement or monitoring unit, the medical device's core which depends on each device's primary function, is detailed. In addition to reviewing the leading technologies, this chapter draws out specific technical requirements that each component must meet to be used in the medical sector. This chapter highlights that the design of embedded systems for the medical industry must comply with specific aspects not required in other sectors.

2.1 Typical Architecture

An embedded medical device usually comprises various hardware and software elements, such as processors, operating systems, memories, power supplies, user interfaces, communication interfaces and other peripherals such as scanners, cameras, buttons or pedals. In the following subsections, the different elements that make up an embedded medical device are described. In addition, the main requirements must be met to include such components in a medical device are extracted. Figure 2.1-1 shows the main components of an embedded medical device and the relationships between each of them.

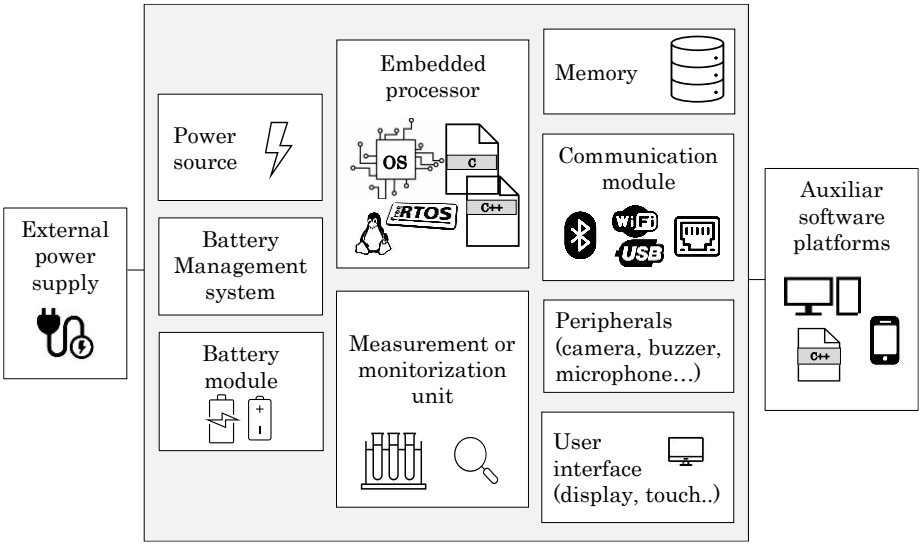


Figure 2.1-1: Typical embedded medical device block diagram.

2.2 Embedded Processors

Embedded processors are hardware elements designed to cope with the needs of embedded systems. They usually have low power and low computing capacity, but they are highly optimised processors. In contrast to traditional processors, they often include built-in peripherals to optimise power consumption and cost further.

Processing performance is measured in units of Dhrystome Millions of Instructions Per Second (DMIPS) [110]. Dhrystome is a computer program developed by Reinhold P. Weicker in 1984. It is used as a

benchmark measurement of the overall performance of different processors. DMIPS refers to the average number of instructions executed per second, so the higher the DMIPS, the higher the processing performance of the processor.

Different data processors are available, including microprocessors (MPU) [111], microcontrollers (MCU) [112], Digital Signal Processors (DSP) [113], System on Chip (SoC) [114] and Field Programmable Gate Arrays (FPGA) [115], which are the typical ones. These are all commonly used in embedded systems; one or a combination of several are used to be the processing core of embedded medical devices.

The processor is critical when designing an embedded system for the medical sector. When selecting a processor for a medical device, it is essential to **ensure that the platform has protection mechanisms**, such as source code encryption, a secure storage system for encryption keys, advanced update systems, etc. [116]. Likewise, suppose an embedded processor will be used in a design on a module-type platform. In that case, it is essential to **identify manufacturers that have ISO 13485** certification or similar. For example, Variscite, a System On Modules (SOMs) manufacturer, produces all of its SOMs in compliance with ISO 13485 [117]. In addition, the processor **supply must be guaranteed for the device's lifetime**. Due to the regulatory requirements, medical devices must pass costly tests to be marketed. A change in a component, such as a processor, can result in the re-certification of the device.

Furthermore, the different embedded processor manufacturers usually offer firmware/software libraries that help to start up the various peripherals. It is also important to check that the **resources offered are designed according to medical device regulations** or that enough information exists to adapt and verify these developments.

2.2.1 Microcontroller (MCU)

A microcontroller is a device that includes the main elements of a computer system inside an integrated circuit. MCUs integrate a CPU and peripherals such as watchdogs, memories, ADCs or PWMs. Typically, on-chip flash memory stores and executes the program, so there are usually space limits [118]. Having on-chip the main

hardware elements simplifies the electronic design making it feasible even for inexperienced engineers.

These devices are size optimised, making them very interesting for small devices with real-time requirements. Due to their small size, they usually do not have large processing capabilities. However, they offer high flexibility and a very short start-up time.

MCUs are used with real-time embedded operating systems. However, only operating systems with very low computing requirements, such as FreeRTOS, QNX or Zephyr, can be used due to their limited computing performance.

Regarding its use, it is mainly used in applications with minimal processing requirements and when there is no need to manage large amounts of data or peripherals. For example, they can be used for electronic lock control, microwaves, or wearables.

Currently, most MCUs are based on ARM architecture processors [119], which have become increasingly popular. Due to the European semiconductor crisis, processors based on the RISC-V [72] architecture have been developed in recent years. These processors are based on a free and open architecture. Any person, institution, or company can freely use and modify them.

Manufacturers such as STMicroelectronics, NXP, Infineon, Nordic Semiconductor or Silicon Labs have MCUs based on ARM processors. Nowadays, the most used MCUs are ARM Cortex M, which are highly oriented to simple, low-cost and low-power applications such as IoT or embedded devices. Table 2.2-1 presents the main characteristics of some families of Cortex M microcontrollers.

Table 2.2-1: ARM Cortex M family comparison [120].

Features	Cortex M0	Cortex M1	Cortex M3	Cortex M4	Cortex M55	Cortex M7
Instruction set	Armv6-M	Armv6-M	Armv7-M	Armv7-M	Armv8.1-M	Armv7-M
Core bit-width	32	32	32	32	32	32
Floating Point Unit	No	No	No	Single	Half, Single Double	Single Double

Features	Cortex M0	Cortex M1	Cortex M3	Cortex M4	Cortex M55	Cortex M7
Digital Signal Processing Extension	No	No	Yes	Yes	Yes	Yes
Instruction /Data cache (kB)	-	-	-	-	64/64	64/64
Instruction/ Data TCM (MB)	-	1/1	-	-	16/16	16/16
DMIPS/ MHz	0.87	0.8	1.25	1.25	1.6	2.14
Core MHz	2.33	1.85	3.34	3.42	4.2	5.01

2.2.2 Microprocessors (MPU)

Microprocessors are integrated circuits that contain several CPUs to process data in parallel. These elements, unlike MCUs, do not have memories or peripherals but are designed to achieve very high processing performance [111].

External NAND or non-volatile flash memory usually provides data and program storage. During start-up, the program is loaded into an external DRAM. Boot times are not as fast as in MCUs, as the program must be dumped between memories during booting.

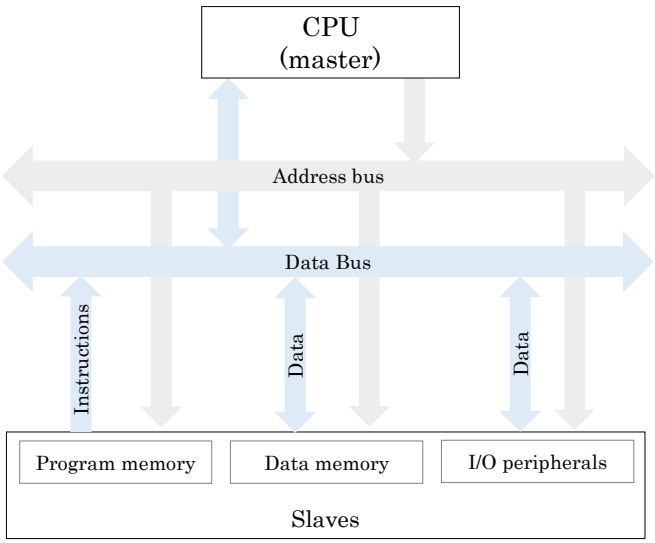
MPUs are widely used with resource-intensive operating systems, e.g. Yocto (Linux), Windows CE, Android, etc. These OSs are usually not suitable for dealing with real-time behaviour. However, they offer plenty of programming resources such as libraries, display drivers, abstraction layers, etc.

Hardware design using MPUs tends to be complex. Firstly, it is necessary to integrate all external components, such as peripherals, memories, etc. Second, MPUs require several power supplies, as each element usually works at a different voltage. Also, the power consumption is generally higher for MPUs than for MCUs, even though they have low-power modes.

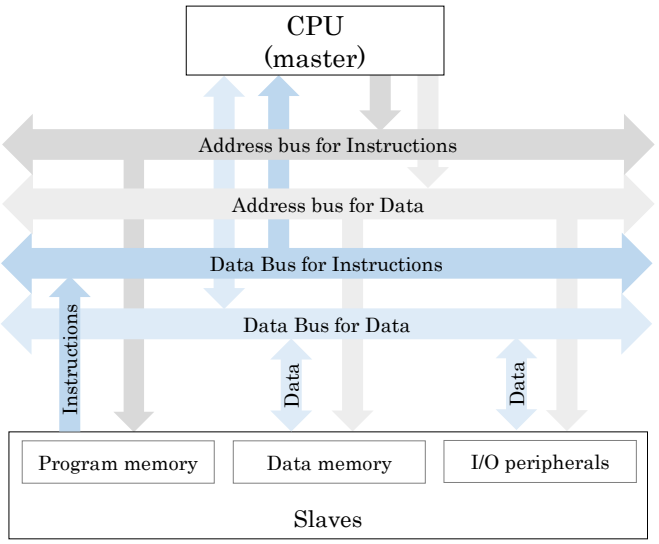
Depending on the data and program storage strategy, the microprocessor can be implemented as a Von Neumann or Harvard architecture [121]. In Von Neumann, instructions and data are

stored in memory and accessed through a single address, data and control bus. In contrast, in Harvard's architecture, the address, data and control buses are independent of the data access buses.

Regarding the applications they are used, it is common to integrate MPUs when multiple communication interfaces, complex HMI designs or extensive data processing are required.



(a)



(b)

Figure 2.2-1: (a) Von Neumann and (b) Harvard architecture.

The most popular MPU families are ARM's Cortex-A MPUs, designed to address compute-intensive applications such as mobile phones and televisions.

Table 2.2-2: ARM Cortex-A family comparison [122].

Features	Cortex A5	Cortex A7	Cortex A9	Cortex A35	Cortex A73
Instruction set	Armv7-A	Armv7-A	Armv7-A	Armv8-A	Armv8-A
Number of cores	1-4	1-8	1-4	1-4	1-4
Core bit-width	32	32	32	32/64	32/64
Floating Point Unit	Yes	Yes	Yes	Yes	Yes
Digital Signal Processing Extension	No	No	Yes	Yes	Yes
L1 Instruction/ data cache	4-64kB	8-64kB	16-64kB	8-64kB	32-64kB
L2 cache	External	1MB	External	1MB	8MB
DMIPS/MHz	1.57	1.9	2.5	2.5	4.8
Core GHz	1	2.2	2	1	2.8

2.2.3 Digital Signal Processor (DSP)

DSPs are a type of microprocessor specifically designed for high-speed digital signal processing [123]. These elements are usually optimised to perform complex signal-processing tasks requiring arithmetic operations involving multiplications, complex numbers, etc. Therefore, they can perform tasks such as digital filtering and frequency analysis. DSPs are designed to perform real-time processing tasks, so they have optimised memory systems and parallel processing capabilities.

The main difference between a DSP and a microprocessor is that the DSP provides features designed to handle high-performance, repetitive and numerically intensive tasks. In contrast, microprocessors are not optimised for any application; they are for general purposes. In terms of design complexity, DSPs are not more complex than microprocessors, as they are a particularisation of microprocessors.

The typical DSP applications are wireless headsets, Bluetooth speakers, radar applications, etc. DSPs are often used when the medical device requires image processing tasks. The leading DSP families currently available include manufacturers such as Texas Instruments [124] or Analog Devices [125].

2.2.4 Field Programmable Gate Array (FPGA)

FPGAs are customisable processing integrated circuits that can be adapted to provide several combinational logic circuits to meet user requirements [126]. Unlike traditional microprocessors, the FPGA is a hardware implementation. Internally, they are built up with logic gates, wires, bi-stables and input and output ports. Using specific programming languages such as VHDL or Verilog, it is possible to describe the desired digital circuit and link all these components according to the application's requirements.

The programmability of FPGAs is their main advantage over microprocessors, making them much more efficient and powerful than general-purpose MPUs. They are especially suitable for applications where many parallel operations need to be executed. On the other hand, programming FPGAs is much more complicated than microprocessor coding, as working with FPGAs requires knowledge of digital electronics. Additionally, hardware design is more complex as it involves the design of several levels of power supply, clocks, memories, etc.

Due to their customisable nature, FPGAs can be used for various tasks and solutions. They are common in specific applications requiring data processing; FPGAs can be found in telecommunications servers, web search engines, high-speed data transmission applications, etc.

Nowadays, Intel [127] and Xilinx [128] stand out as FPGA manufacturers; they have many FPGA families and resources for designing and programming these devices.

2.2.5 System on Chip (SoC)

The SoC is a hardware device that integrates most components required to design and operate an embedded system [129].

Microprocessors, connectivity modules, DSPs, display drivers, security modules, etc., used to be integrated into a SoC.

This device has the same nature as an MCU. Still, a SoC has more specific peripherals, is focused on applications with higher computing requirements, is more expensive, offers much more memory and has a higher power consumption. SoCs can include other processors such as MPUs, FPGAs or MCUs.

In terms of applications, SoCs are widely used in portable devices that require high computing capacity, such as mobile phones. Currently, due to the high demand for these integrated circuits, most of the processor manufacturers have SoCs, the most important of which are NXP [130], Intel [131], Xilinx [132] or Qualcomm [133].

2.3 Embedded Operating System (OS)

The operating system is a set of programs that allows the management of hardware resources available on an embedded system [134]. Embedded OSs often rely on platforms with limited processing capacity, so they tend to be highly optimised. Unlike desktop operating systems such as Windows 10 or Debian, they generally do not offer many software resources and tools. In most cases, the developer has to generate such customised libraries.

Embedded operating systems' main characteristics are the multi-tasking capability, **real-time operation and suitability for safety-critical applications**, such as medical devices or automotive solutions. Some operating systems are **pre-certified for critical applications**, easing the embedded device's certification process. In addition, the operating system to be used depends on the platform on which it will be executed [135].

Currently, some of the operating systems that can be found pre-certified for medical devices are Wind River [136], Integrity [137], QNX [138], Nucleus [139] or SafeRTOS [140]. They all offer software and documentation ready to be integrated into medical devices.

In the case of operating systems that are not certified for the medical sector, it is necessary to use operating systems that allow the separation of different parts of the software. For mitigating or

simplifying risks during the design and development phase of medical devices, it will be essential to separate safety-critical and non-critical software [141]. Critical software is the one whose malfunction could harm the patient or operator.

Furthermore, for medical systems that require real-time response, typically systems that generate alarms or are capable of dosing medication, it will be necessary to ensure that the operating system offers very low response times.

2.3.1 OS for Microcontroller

Microcontroller operating systems are usually oriented to run real-time applications. They are designed to require minimal processing resources to avoid overloading the MCU with a high computing load[135]. Furthermore, they offer few programming resources, essentially a kernel that provides facilities for scheduling, implementation of communication tasks, timers, and interrupt management. It also provides drivers to easily access peripherals such as I2C, SPI or I/O.

Using an operating system is not always required; programming MCUs without an OS is possible if the application is simple. Systems that do not integrate an operating system are known as bare-metal. Some MCU OSs are Zephyr, VxWorks, SafeRTOS/FreeRTOS and Nucleus. All offer resources for task management, semaphores, timers, and drivers to access peripherals. In addition, VxWorks, SafeRTOS or Nucleus have specific distributions already pre-certified for medical devices.

2.3.2 OS for microprocessors

Microprocessor OS typically require more processing resources than microcontroller OSs. By their nature, they are not always geared to meet real-time requirements.

MCU OS usually offer several tools and libraries for accessing and controlling the hardware elements. It is common to find OSs with resources such as graphic design tools, communications monitoring, drivers for display controllers and memory controllers. Thanks to their many tools, the programmer can design the software with the maximum hardware abstraction level.

Typically, operating systems for embedded MPUs are based on desktop OSs, such as Linux or Windows. Some OSs based on Linux are Yocto, Wind River Linux, Sokol Flex OS, Emlix, Android and Integrity. There are also some Windows-based OSs, such as Windows for IoT and Windows CE.

Table 2.3-1 compares the most relevant OS for microprocessors and microcontrollers.

Table 2.3-1: Medical OS comparison.

Characteristics	SafeRTOS [140]	Nucleus [139]	Sokol Flex OS [143]	QNX [139]	Integrity [137]	Wind River Linux [142]
Core support	Cortex-M, Cortex-A, ARMv11, ARMv9, Power Architecture, Power PC	Cortex-A, Cortex-R, Cortex-M, ARMv8	Cortex-A, PowerPC, AMD x86, Intel x86	Cortex-A, Cortex-M, Snapdragon, AMD x86, Intel Core i	Cortex-A, Cortex M, Snapdragon, AMD x86, Intel Core i	Cortex-A, Intel x86
Real-Time OS	Yes	Yes	No	Yes	Yes	No
Communications	USB	USB	USB, WiFi, ETH, BLE	USB,	USB, WiFi, ETH, BLE	USB, WiFi, ETH, BLE
FFS, FAT and NOR/NAND Flash support	Yes	Yes	Yes	Yes	Yes	Yes
Fault control mechanism	No	Yes	Yes	Yes	Yes	Yes
Security on communications	No	Yes	Yes	Yes	Yes	Yes
Pre-certificated for FDA and CE	Yes	Yes	No formally, references for class I and IIa	Yes	Yes	Yes

2.4 Memory

The memory or data storage device stores the embedded program, firmware, and user data. It can be classified into volatile [144] and non-volatile memories [145], [146].

In a medical device, data storage memories are especially important. These elements must contain measurement or monitorization algorithms, device critical algorithms and sensitive data.

The main characteristics of memory devices are the following:

- Nominal capacity: The amount of information (bytes) a device can store.
- Access speed or access time: Time elapsed from the moment that the memory address is provided until the data is available.
- Memory cycle: Time elapsed between two consecutive memory accesses.
- Cost per bit: Price per information bit.
- Volatility: This parameter indicates whether the stored information is lost when the power supply is cut off.
- Lifetime: Some memories have a limited number of write cycles. Therefore, this parameter is defined as the number of write cycles performed without the risk of losing information.

Similarly to embedded processors, manufacturers of medical memories must have **ISO 13485 manufacturing certification**, and their storage devices must be certified. Memories used in the consumer market are generally not suitable for medical devices. These elements must offer **long-term availability** to avoid possible re-certification of the product. In addition, **components with high reliability and security features** should be chosen.

Medical device memories often contain sensitive information, so the data must be guaranteed inviolable and can only be read by authorised personnel. This problem is usually solved through **encryption mechanisms** and **access control** with authentication.

2.4.1 Volatile Memory

Volatile memories are Random Access Memories (RAMs) [144]. RAM stores data and programs that need to be accessed very quickly. This type of memory is volatile, meaning the data is only maintained as long as the device's power supply is maintained. There are different types of RAMs, such as DRAM, SDRAM and DDR RAM.

In Dynamic RAM (DRAM), the data is stored in capacitors, so although the power supply is maintained, it is necessary to refresh the data not to be lost periodically. DRAMs generally offer large storage capacities at a very low cost; however, data access is often slow compared to other technologies, and power consumption is typically high.

A variation of DRAM is Synchronous DRAM (SDRAM). These memories are optimised according to the microprocessor's clock, so access is performed synchronously. In this way, achieving better efficiency for storing and accessing the stored information is possible.

There are also Double Data Rate RAMs (DDR RAMs) which are an evolution of SDRAMs. These memories can perform two data transfers per clock cycle, whereas an SDRAM can only perform one.

Static RAM (SRAM) uses transistors to store data. Due to the used configuration, there is no energy loss, so if there is a power supply, it is unnecessary to refresh them periodically. They are usually used as cache memories. SRAMs also consume less power and have faster access speed than DRAMs. On the other hand, the memory size is generally smaller than in DRAM, and the manufacturing costs are higher.

2.4.2 Non-volatile Memory

Non-volatile memories are Read Only Memories (ROMs) [146]. These memories keep the data stored despite not being powered up. They typically store code and firmware data that will not be modified continuously. There are several types of ROMs, including PROM, EEPROM, Flash Memory and SD Card.

Programmable ROMs (PROMs) are memories in which information is recorded only once; this information remains fixed forever and cannot be deleted or modified.

Erasable Programmable ROMs (EPROMs) are memories that can be programmed and erased by ultra-violet light. Deleting all stored data and re-recording it to reprogram the device is necessary.

Electrically Erasable Programmable ROMs (EEPROMs) can be erased and programmed using an electrical voltage. In addition, only some parts of these memories can be erased and programmed, offering high flexibility to the programmer. Despite this, the storage process is relatively slow.

Flash memories are a type of EEPROM memory that can be rewritten considerably faster than EPPROMs as it can operate with large blocks of memory. Also, Flash memories support a higher number of write cycles. For example, SD cards or USB flash sticks integrate Flash memories.

2.5 Battery Powered Devices

Medical devices can be stand-alone (battery-powered) or plug-in devices that must be continuously connected to an external power source. Battery-powered devices are typical in wearable monitoring applications or PoC (Point of Care) devices. Plug-in devices are used mainly in energy-intensive equipment.

Low-capacity lithium batteries are usually used to power battery-powered devices. These batteries rely primarily on the movement of lithium ions between the positive and negative electrodes of which they are composed [147]. The negative electrode material is fuel, so if the battery is overcharged, there is a potential risk of explosion. Also, these cells dissipate heat in a non-uniform way, which can considerably reduce the battery's lifetime. For this reason, lithium batteries need to have a monitoring, management, and protection system.

When selecting a battery, it is necessary to consider the following parameters:

- Nominal capacity [148]: This parameter indicates the amount of discharged electricity from the battery under certain conditions, such as discharge rate and temperature. This capacity is the maximum power the battery can supply per hour.
- Battery charge and discharge rate [149]: It indicates the speed at which the battery is charged and discharged. It is defined as the charge or discharge current divided by the battery's nominal capacity. For example, if a 4000 mAh battery is discharged at 1000 mA, its discharge rate is 0.25C.
- Depth of Discharge (DOD) [150]: This parameter refers to the battery depletion rate. It indicates the percentage of the discharged capacity concerning the nominal rating during use.
- State of Charge (SOC) [151]: It indicates the percentage of the remaining energy compared to the battery's nominal capacity.
- State of Health (SOH) [151]: It refers to the battery's state of health. This parameter evaluates the state of the battery compared to the ideal conditions. It is usually represented as a percentage value.
- Internal battery resistance [152]: Batteries have an internal resistance as the elements that compose them are imperfect conductors. This parameter will vary with age. The higher the resistance, the higher the energy losses; therefore, the higher the battery will heat up.
- Life cycles [153]: It refers to the number of charge and discharge cycles a battery can undergo before its capacity drops below a specific value. This parameter will depend on the quality and the materials of the battery.

The Battery Management System (BMS) [154] is the batteries' control system. The BMS estimates battery charge, monitors and analyses battery health, implements safety mechanisms, and manages energy control.

Usually, the BMS is composed of several electronic components. It is essential to know the main blocks forming the base of the device's power supply system to design a battery-powered medical device.

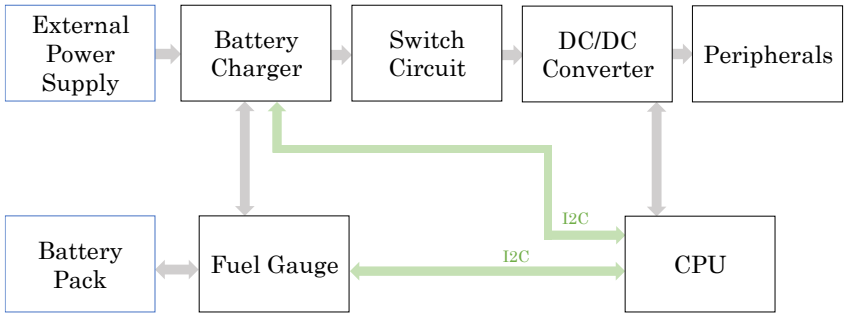


Figure 2.5-1: Battery Management System block diagram.

The figure shows the main elements that compose the power supply system [155]–[159]:

- Power supply: It is the external power input and is typically designed to connect an external power adapter.
- Battery Charger and Fuel Gauge: They are responsible for setting the charging conditions and counting the battery's charge level. A power path architecture is often used where the system can be powered from either the input voltage or the battery. These devices usually have I2C communication to access configuration and monitoring registers.
- DC/DC: It regulates the voltage coming from the battery or from the external input for powering the processor and peripherals. Its operation is conditioned by the signal generated by the switching circuit. In idle conditions, the DC/DC converter is inactive and does not supply voltage to the module. In this way, power consumption can be minimised.
- Switching Controller: It activates and deactivates the DC/DC converter. The on and off condition is typically generated by pressing the power on button.
- CPU module: This is the module where the application program is executed. When idle, it has no power supply. It will start operating when there is power because of the abovementioned conditions. This element can usually access the Charger and Gauge registers via I2C, which allows it to characterize and control its operation. Likewise, it can turn itself off by acting directly on the DC/DC.

- Battery: It is the energy storage unit.

Medical devices may include primary or rechargeable batteries. This **element** is **considered critical** as it can overheat and explode, **causing harm to the patient or operator** [160]. In the case of batteries, standards such as UL 2054 or IEC 62133 must be met when integrating a battery into a medical device. These standards will be discussed in section 3.10.

In addition to **integrating pre-certified batteries**, it is necessary to **take design measures to protect the system**. For example, the **limitation of the maximum charging current** both by hardware and software and **integrating redundant temperature sensors** that monitor the battery status.

2.6 User Interface

The user interface is the set of peripherals and channels through which a user can communicate with a medical device. Typically, medical devices include displays, touch panels and buttons to interact with the end user.

Medical grade display or touch screens must be **certified** according to IEC 60601-1. In particular, when choosing displays for use in the medical sector, it is essential to check **technical characteristics** such as **usability with gloves, viewing angles or luminance**.

Luminance (screen brightness) and contrast (light-to-dark ratio) are critical. A **wrong display selection can lead to an incorrect diagnosis** if the screen is intended to be the diagnostic platform. A display for diagnostic purposes must offer enough luminance and contrast to create at least 256 visually perceptible shades of grey [161].

Touch screens are often used while wearing gloves, so the selected device must support this use case.

Finally, the viewing angle is another critical parameter; it defines the maximum angle through which a good screen view is obtained. Often the operators view the screen from different positions as they move around the surgery or examination room. Therefore, it is crucial to **choose screens with In-Plane-Switching (IPS) technology**

that maintains a good ratio of luminance and contrast at high viewing angles [162].

2.6.1 Screen

Many types of displays can be integrated into embedded devices; the most used are TFT LCDs, PMOLEDs, AMOLEDs and electronic ink displays. The most important characteristics used to compare different displays technologies, besides the usual ones such as cost or power consumption, are the following:

- Readability: The ease of reading the display in different environments, for example, where there is high or low ambient light.
- Contrast: The difference in colour that a screen has between a white pixel and a black pixel.
- Viewing angle: The ability of a display to show a viewable image from different angles. The higher the viewing angle, the better the screen will be seen from any angle.

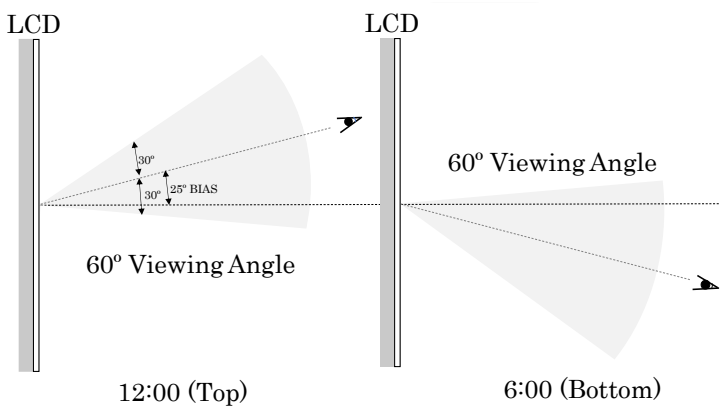


Figure 2.6-1: Display viewing angle.

- Colour reproduction: The ability of the display to present different colours.
- Update speed: The speed at which the entire screen is refreshed. A low update speed can cause the user to appreciate how the screen is erased and painted over.

Thin Film Transistor Liquid Crystal Displays (TFT LCDs) [163] are among the most widely used displays. Thanks to a backlight system,

they have good visibility in darkness. In addition, the screen's refresh rate is usually very high; they have good colour reproduction and high durability. Screens are available in various sizes, and their cost is moderate.

Active Matrix Organic Light Emitting Diode (AMOLED) [164] displays have excellent image quality and readability in darkness. They also offer good durability, viewing angle and power consumption. On the other hand, their cost is very high.

Passive Matrix Organic Light Emitting Diode (PMOLED) [165] are cheaper than AMOLEDs and have a better viewing angle. However, they require more current to paint the image and have a slower refresh rate. The higher current requirements can lead to faster degradation of the displays. Like AMOLED, they are suitable for use in the darkness and present a high contrast. They are typically used for small screen sizes, and their cost is moderate.

Electronic ink displays [166] are usually monochrome and have perfect viewing angles, durability, and readability under bright light. Their power consumption and cost are very low, making them suitable for small battery-powered devices. On the other hand, they do not provide good readability in darkness and have a long refresh time.

2.6.2 Touch panel

It is more and more common for medical devices to use touch panels. They allow users to interact with the devices simply and comfortably. These panels have sensors that detect the user's finger actions and then translate them into specific instructions. There are several types, but capacitive and resistive systems are the most widely used for embedded systems.

2.6.2.1 Resistive touch panel

Resistive panels detect the pressure applied to the panel's surface [167]. They usually have several layers of polycarbonate and glass. These layers are typically covered with conductive and resistive material. An air gap is left between the different layers. At the location where the touch occurs, the air gap is removed. The two layers eventually contact each other, so it is possible to detect in

which region the panel's resistance has changed and determine the position of the touch.

These panels are accurate and especially good when used in harsh environments. These panels are usually not multipoint; they only detect the variation of pressure at a single point simultaneously. On the other hand, they present high noise immunity, are suitable to be used with gloves and pencils, and are very resistant to noise. They also have an increased lifetime and low manufacturing costs.

They are commonly used in environments where accuracy is essential and where they face possible interferences generated by elements such as water, dirt or different contaminants. They are primarily used by hospitals, industry, video games or even in virtual reality systems.

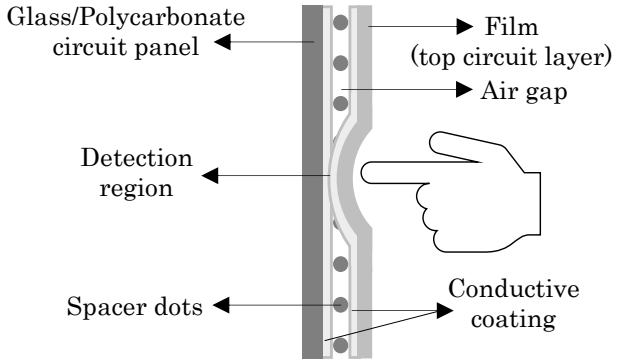


Figure 2.6-2: Resistive touch panel structure.

2.6.2.2 Capacitive touch panel

This type of touch panel uses a capacitive sensor [168]. It is formed by a layer of electrical insulation covered with a transparent conductor. Often these layers are then covered with glass. Several electrodes are placed on the panel to maintain the voltage level in the conductive layer. As the human body is conductive, a current flow is generated between the electrodes and the finger when pressing on it. In this way, screen sensors can detect voltage variations and therefore identify the location of the touch.

Capacitive panels have higher sensitivity than resistive panels and can detect much weaker touches with higher accuracy. Also, this higher sensitivity makes it possible to have multi-touch screens.

Capacitive panels are the best choice in applications requiring high contrast and brightness. Because of their number of layers, resistive displays tend to be affected by higher reflections. In terms of cost, capacitive touches are more expensive to manufacture than resistive touches. Finally, as they have the multi-point capability, they are used in applications requiring sophisticated actions, e.g., mobile phones and computers.

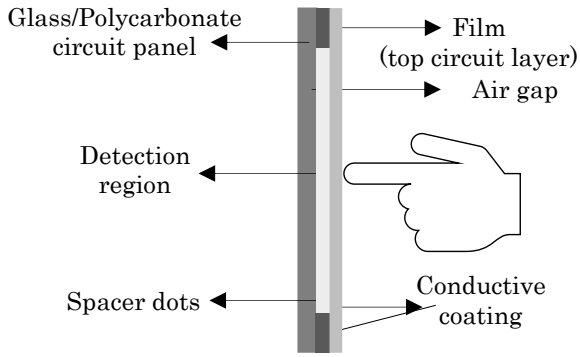


Figure 2.6-3: Capacitive touch panel structure.

2.6.3 Haptic systems

Haptic interfaces allow the user to feel their actions [169]. That is, the user receives feedback through the touch panel in a non-visual and non-audible way.

In practice, haptic systems allow the user to receive feedback from the device as a force. They can touch, feel, and manipulate objects using the sense of touch. There are several types of haptic systems, for example, based on piezoelectric actuators, ultrasound, or concentrated air.

The most widespread haptic systems are those based on piezoelectric. When the piezo effect [170] occurs, a vibration is generated when an electric current stresses some materials. The main elements of this type of haptic system are a display, a capacitive touch panel, a processor, and a piezoelectric actuator. The

touch panel identifies the position in which the user places the finger; this data is processed in a microprocessor which activates the piezoelectric actuator as a feedback signal to the user.

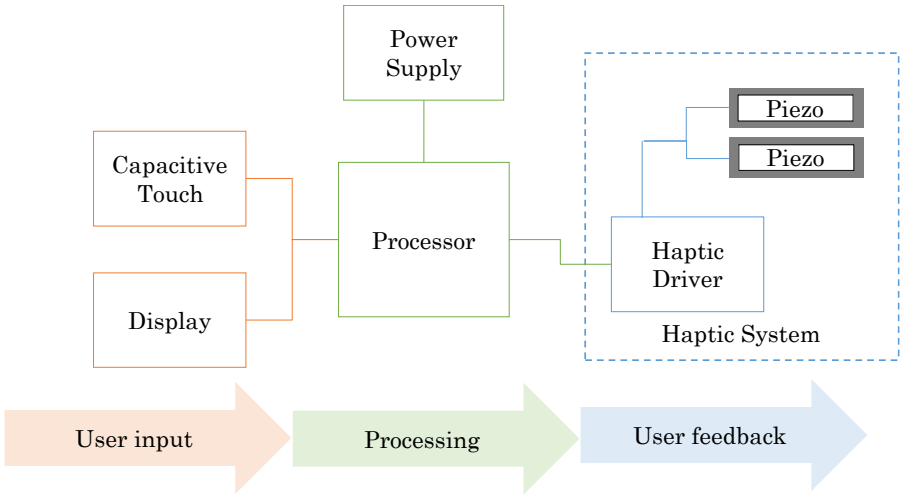


Figure 2.6-4: Haptic system block diagram.

Although these systems are not widespread, they could be used in many applications. Robots for surgery, design of industrial components, virtual exhibitions or video games could be some of them.

2.7 Communications

The interoperable exchange of information provides significant benefits for healthcare systems and telemedicine. On the one hand, it makes it easier for healthcare providers to have all patient data centrally available. On the other hand, patients have access to much information about their health status.

The exchange of information requires interoperability between different medical devices as well as between data servers. Therefore, **standardising communication protocols is an essential** requirement for new e-Health solutions. In this context, to overcome integration difficulties arising from the lack of standard communication interfaces and communication homogeneity, it is necessary to **use standard protocols** such as those defined in ISO/IEEE11073 (X73), HL7 or POCT1-A2.

2.7.1 Health Level Seven (HL7)

Health Level Seven (HL7) is a communication standard that provides resources for information exchange between different healthcare information systems. This protocol is defined and maintained by Health Level Seven International, a not-for-profit organisation working to create standards for healthcare. This organisation has a membership of more than 500 healthcare organisations in over 50 countries [171].

HL7 aims to achieve interoperability between different hospital systems. This protocol is used in hospitals, medical centres, laboratories, pharmacies, emergency services, medical hardware and software manufacturers, etc. The HL7 standards consist of 5 groups: HL7 Version 2, HL7 Version 3, CDA, HL7 FHIR and CCOW.

The HL7 Version 2 (V2) [172] standard defines the exchange of clinical information between hospitals and medical centres. This release is currently the most widely used and is supported by most healthcare systems. It is very flexible and generalist, so its integration often requires a lot of negotiation and analysis between the parties.

The HL7 Version 3 (V3) [173] standard was defined to specify further the terms, data format and messages necessary to complete an interoperable communication with fewer integration challenges than its predecessor, HL7 V2. HL7 V3 is based on the Reference Information Model (RIM), a healthcare information model.

The HL7 Clinical Document Architecture (CDA) [174] defines the interoperability between different types of clinical information. It mainly works on the structural and semantic parts of medical documents. This standard is also based on the RIM data model and the architecture defined in HL7 V3.

HL7 Fast Healthcare Interoperability Resources (FHIR) [175] is an interoperability standard that combines HL7 V2, V3 and CDA. This standard defines the information exchange blocks intending to be able to model clinical procedures quickly.

The HL7 Clinical Context Object Workgroup (CCOW) [176] aims to facilitate application integration at the user level by unifying user interfaces from different systems.

All these HL7 standards are specially designed to exchange large amounts of data where limited processing resources are not an issue. Despite being one of the most widely used standards worldwide, it is currently not a standard intended to be integrated into resource-limited embedded devices.

2.7.2 ISO/IEEE 11073

ISO/IEEE 11073 (X73) is a family of standards that aims to ensure interoperability between different medical devices [177]. For this purpose, there are two groups of standards Point-of-Care medical device (PoC-MDC) and Personal Health Device (PHD).

The X73 family of standards covers the seven levels of the OSI protocol stack, providing flexibility for exchanging medical data between a Medical Device (MD) and a central system, the Compute Engine (CE). In addition, the data exchanged between the MD and the CE can be sent to a remote-control centre for storage in the Electronic Health Record (EHR). Communication with the EHR is regulated by ISO 13606 [178].

The X73 standard was created to be implemented in Intensive Care Units, specifically for Point-Of-Care (PoC) devices. Subsequently, it has undergone several evolutions towards new use cases and profiles. Accordingly, the European Committee for Standardisation has adapted the X73 to the current technological scenario with an extension for personal healthcare (Personal Health Device, X73PHD).

The IEEE 11073 PHD defines the format of the messages that will travel between the agent and the manager. However, it does not determine how the messages have to travel. This standard currently supports three transport layers Bluetooth, USB and Ziggbe.

The X73 standard is currently defined for most medical devices that require a specific profile to communicate via Bluetooth Classic. Although X73 does not directly support Bluetooth Low Energy (BLE), it supports profiles and services compatible with compliant

data types. In this way, it is possible to transcode them into equivalent X73 representations using software services such as mobile phone or tablet applications.

It is possible to find multiple examples and guidelines for this type of BLE device integration developments in X73 ecosystems. Even the Medical Device Working Group has published a white paper on BLE transcoding to X73. Custom profiles can also be defined for applications without already defined profiles in the standard. The specifications currently available are Blood Pressure Profile (BLP) [185], Continuous Glucose Monitoring Profile (CGMP) [186], Glucose Profile (GLP) [187], Heart Rate Profile (HRP) [188], Health Thermometer Profile (HTP) [189], Pulse Oximeter Profile (PLXP) [190], Weight Scale Profile (WSP) [191] and Insulin Delivery Profile (IPD) [192].

In X73, the information transfer occurs between the "Agent" and "Manager". An "Agent" is a software component within a Personal Health Device that provides a functional interface to devices in the outside world. The "Manager" is also a software component on the other side of the transport channel, and it receives data from the agent.

The PoC MDC refers to communication standards for Point-of-Cares and defines the seven layers of the OSI stack. The best-known standards are the following:

- ISO/IEEE 11073-10101: Health informatics – Point-of-care medical device communication – Part 10101: Nomenclature [179]
- ISO/IEEE 11073-10201: Health informatics – Point-of-care medical device communication – Part 10201: Domain information model [180]
- ISO/IEEE 11073-20101: Health informatics – Point-of-care medical device communication – Part 20101: Application profile – Base standard [181]
- ISO/IEEE 11073-30200: Health informatics – Point-of-care medical device communication – Part 30200: Transport profile – Cable connected [182]

The data model defined in PHD is divided into two main components [183], [184]:

- DIM (Domain Information Model):

It is a medical information tree. It describes a set of atomic parts and how they are grouped to form a larger element that could be used in the software.

- Service Model:

It describes a protocol for interacting with objects and attributes. There are three types of components: association services, object access services and communication model.

The association service establishes and releases a logical connection. The object access service is used to access the attributes within the information objects defined in the DIM. The supported services are "get", "set", "event report" and "action service", by which it is possible to collect and write information and request and send information and data.

Finally, the communication model describes the connection state machine and the communication's characteristics.

2.7.3 Point-of-Care Connectivity (POCT1-A2)

POCT1-A2 is an optimised communication standard for Point of Care (PoC) devices [193]. POC are medical devices used to obtain results quickly, without needing a laboratory. Medical devices such as glucometers, coagulometers or thermometers are considered PoC devices. These are usually embedded systems with limited resources.

In 1990, the American Society for Testing and Materials (ASTM) [194] published the first specification for exchanging information between medical instruments and computer systems (ASTM E1394). This guideline was not widely deployed, so in 2002, the Clinical and Laboratory Standards Institute (CLSI) defined the Clinical Instruments and Computer Systems (LIS02-A2) [195], which replaced the previous one. LIS02-A2 is currently known as POCT1-A2. This standard was accepted by the International

Organization for Standardization (ISO) and is covered by ISO11073-90101 [196].

The POCT1-A2 protocol describes four typologies of elements located at different levels of the communication architecture:

- Point-of-Care Device(s): It is the device or environment that monitors the patient.
- Access Points (AP): It is the transport layer router or adapter. It is used when the device cannot connect directly to the Observation Reviewer, for example, because it communicates via infrared.
- Observation Reviewer (OREV): It is the server that stores, processes, and manages data from PoC devices. The OREV performs standard support tasks such as test result management, quality control and medical orders.
- Observation Recipient (OREC): It is the hospital and laboratory database, also known as Clinical Information System (CIS) or clinical database known as Healthcare Information System (HIS). Specific examples include Laboratory Information Systems (LIS), Clinical Data Repositories (CDR), and Electronic Medical Records (EMR).

The standard also defines two communication interfaces, the "Device Interface" between the PoC device and OREV and the "Observation Reporting Interface" between OREV and OREC. In addition, within these two interfaces, three communication protocols are specified:

- Device and Access Point (DAP):

It is the DML's low layer specifying the protocol between the PoC device and the access point or gateway.

The standard specifies the DAP protocol for communication between PoC devices and APs. However, this protocol is oriented towards serial and infrared channels. Although mentioned, it does not provide information about its use on other channels, such as Bluetooth. TCP/IP, the default protocol in Ethernet and Wi-Fi, is a valid transport protocol.

- Device Messaging Layer (DML):

DML is the high-level communication protocol between the medical device and the OREV. It is based on the XML text format, and by using XML-based data types, there is virtually no limit to the data size.

DML does not implement any transport management mechanism, so it must be implemented on top of a transport protocol that is robust and reliable. That is, the transport layer must ensure the integrity of messages and indicate if messages have reached or not the destination. Therefore, it does not implement mechanisms such as retries or CRCs. However, it provides some tools for the application's error handling. These include ACKs, timeouts, single message identification and error notification.

- Observation Reporting Interface (ORI):

The high-level protocol to communicate between OREV and OREC.

2.7.4 Communication interfaces

Regarding communication interfaces, typical medical embedded device communication interfaces include Ethernet, Wi-Fi, USB and Bluetooth.

- Ethernet:

It is defined under the IEEE 802.3 standard [197] as the most used communication interface since 1980. It is also used in healthcare to connect equipment and devices in a local network. Its ability to handle large amounts of data at high speed makes it suitable for providing medical devices with the information they require about patients and results.

Ethernet specification uses an open protocol at the application layer, meaning that standard protocols such as HL7 or POCT1-A2 can be used over this interface.

- Wi-Fi:

It is defined as the standard 802.11 [198] and is increasingly used in hospital environments. Its flexibility, especially in portable equipment, replaces the Ethernet interface in many medical devices.

Wi-Fi allows open protocol over the application layer and can be used with standards such as POCT1-A2 or HL7.

- USB:

USB is also used to communicate with PoC-type medical devices. Usually, the USB interface is used to implement customised and non-standardised protocols according to the customer's needs.

USB is also used when the hospital or medical centre does not have adequate infrastructure to connect the devices to the shared network.

- Bluetooth:

This communication, standard 802.15.1 [199], enables the communication between devices via a secure and globally free radio frequency link (2.4 GHz). Bluetooth is mainly focused on mobile devices and devices where a short to medium range and low power consumption are required. The Bluetooth standard defines two variants intended to be used for various purposes, Bluetooth Classic (BC) and Bluetooth Low Energy (BLE).

Bluetooth Classic can communicate large amounts of data, but its energy consumption is high.

Bluetooth Low Energy (BLE): It is used for applications that do not require massive data exchange and are typically battery-powered. The reduced number of data to be exchanged allows battery-powered devices to communicate with low power consumption.

The two different Bluetooth technologies are not compatible with each other. However, it is possible for a single device to support both variants. BLE was introduced in version 4.0 of the Bluetooth Classic standard as a subset, with a completely new protocol stack to enable the development of simple links quickly.

Both BC and BLE operate in the 2.4 GHz ISM band. However, unlike BC, BLE spends more time in standby mode than sending information. When information needs to be sent, BLE establishes the connection, sends the data, closes the link, and returns to sleep mode within a few milliseconds. Bluetooth was initially designed for applications with continuous data streams. However, many modern devices only need to send data occasionally. For example, a thermometer does not need to send the temperature every millisecond; it is enough to send it once every second.

Since the introduction of BLE in 2011, the standard has continued to receive revisions and improvements. A significant change occurred in 2016 when version 5.0 significantly increased the range, speed, and data capacity. Subsequent versions up to 2020 have introduced, among others, device location and audio-sending capabilities. The evolution of BLE evidence that the current trend in wearable medical device connectivity has been towards using BLE and not the classical BLE variant [200]–[203].

Regarding security aspects, Bluetooth devices, once connected, can perform a "pairing" process, where a series of keys are exchanged to establish future communication using encryption. The pairing process is mandatory in Bluetooth Classic but optional in BLE [204].

Table 2.7-1: Bluetooth Classic and Bluetooth Low Energy comparison.

Specifications	BC	BLE 4.2	BLE 5.2
Range	100 m	> 100 m	> 400 m
Data rate	1 - 3 Mbps	1 Mbps	2 Mbps
Frequency	2.4 GHz	2.4 GHz	2.4 GHz
Latency	100 ms	6 ms	6 ms
Lag	100 ms	3 ms	3 ms
Voice Capable	Yes	No	Yes
Localization Capable	No	No	Yes
Network Topology	Star	Star	Star / Mesh
Power Consumption	1 W	0.01 a 0.5 W	< BLE 4.2

2.8 Measurement or Monitorization Unit

Embedded medical devices, specifically PoC devices, are usually oriented to monitor or characterise some parameters related to patients' health. The functional block associated with this functionality is the measurement or monitoring unit. This module is considered **one of the device's most critical components**. Usually, this unit comprises sensor devices such as temperature, current, voltage, pressure and flow and vision systems like scanners and cameras.

This component is highly **dependent on the intended use** of the medical device to be designed. As discussed previously, there are different measurement principles for various applications. Specifically, the measurement methods associated with monitoring or measuring vital signs have been reviewed in section 1.3.

This system's design implies analysing the different measurement principles to measure or monitor the required parameter. The analysis of the state-of-the-art will help to select the most accurate and appropriate measurement method depending on the device to be developed. Likewise, through this analysis, **scientific evidence will be obtained to prove that the device's design has been carried out based on previously validated scientific trials**.

The monitoring and measurement subsystem is often the **most challenging to verify**. Its validation often requires testing on animals or patients, which is usually very time-consuming and costly. Therefore, this system must be designed to **allow complete control of the progress of the monitoring or measurement process**. It is common to **include different Key Performance Indicators (KPIs)** that continuously evaluate these sub-modules' good performance.

2.9 Technical Requirements for Medical Device Design and Development

After reviewing the main components that make up the typical embedded system, the main requirements to be met are extracted.

In terms of hardware, the use of pre-certified components that ease obtaining the CE marking is identified as critical. Likewise, the need to introduce mechanisms to guarantee the safety of the different components is also highlighted.

Regarding software, it is essential to choose libraries and operating systems with tools for implementing mechanisms to guarantee the security and safety of the device. Finally, it is recommended to use pre-certified operating systems in those processors that contain critical tasks, i.e., those tasks that can cause harm to the patient or operator or whose functions can lead to erroneous data or diagnoses.

Techn. Des. Req. 1	The hardware platform must include protection mechanisms to guarantee the safety and security of the device.
Techn. Des. Req. 2	The hardware component o module manufacturer must have ISO 13485 certification.
Techn. Des. Req. 3	Selected hardware components must be available for the device's lifetime (or alternative components must be available).
Techn. Des. Req. 4	Use pre-certify components (memories, batteries, HMIs etc.) to ease the certification process of the device.
Techn. Des. Req. 5	Software resources such as drivers provided by the processor manufacturer must be suitable for certification as part of a medical device.
Techn. Des. Req. 6	The operating system must provide the required features for the specific application (multi-tasking, real-time extensions, safety and security tools, etc.)
Techn. Des. Req. 7	The operating system must be certifiable as part of a medical device (consider using pre-certified operating systems).
Techn. Des. Req. 8	Identify the device's intended use, and consider all use cases to select the most appropriate

technology. Typical considerations include use in the professional environment (doctor's surgery, operating theatre, ambulance) and home environment.

Techn. Des. Req. 9 Components that can cause harm to the patients or operators are considered critical and must be effectively monitored and controlled to minimise the risk they entail. For example, limiting the maximum current of the device, having redundant temperature sensors in the battery, limiting the use of the device when it is connected to an unstable power source or selecting screens with high luminance, contrast and viewing angle.

Techn. Des. Req. 10 Use standard communication protocols to ensure interoperability between different devices,

Techn. Des. Req. 11 Integrate KPIs into critical processes to ease the verification of each component.

3.
European Medical Device Regulation

During the design and development phases of medical devices, compliance with the standards that regulate these phases is crucial. Therefore, this section aims to analyse the regulations covering the design and development of medical devices in Europe. The main regulations identified are the Medical Device Regulation (MDR) and the In Vitro Medical Device Regulation (IVDR), which determine medical devices' requirements for CE marking. The main standards that apply to the embedded device typology defined in the previous chapter are also described: IEC 60601 (Basic safety and essential performance), IEC 62304 (Medical device software lifecycle), ISO 14971 (Medical device risk management), IEC 62366 (Medical device usability), ISO 13485 (Medical device quality management systems), IEC 81001-5-1 (Medical device cybersecurity) and several battery regulations. For all these regulations, the structure of the standard, its scope and its main processes or clauses are described. Additionally, the regulatory requirements that must be fulfilled during the design and development phases of the medical device according to each standard are extracted.

3.1 Embedded Medical Device Regulatory Requirements

The commercialisation of medical devices in Europe until 25 May 2021 was regulated by three directives 90/385/EEC on Active Implantable Medical Devices (AIMD) [205], 93/42/EEC on Medical Devices (MD) [206] and 98/79/EC on In-Vitro Diagnostic Medical Devices (IVDMD) [207].

From 26 May 2022, two regulations covering medical device commercialisation entered into force. 2017/745 [208], Medical Device Regulation (MDR) and 2017/746 [209], In vitro Diagnostic Medical Device Regulation (IVDR).

In practice, the MDR and IVDR merely lay down minimum requirements that medical devices placed on the European market must meet. The technical details behind these directives are set out in harmonised standards developed by standardisation organisations. Following these European standards, defined in the Official Journal of the European Union (OJEU) [210], confers a presumed product conformity within the legal requirements the standard aims to cover. Despite this, the use of many of the harmonised standards is optional.

According to the MDR and the IVDR, the main harmonised standards that apply to the design and development phase of an embedded medical device are the following:

- IEC 60601 – Medical electrical equipment.
    - Part 1: General requirements for basic safety and essential performance [210].
    - Part 2: Particular requirements for basic safety and essential performance.
    - Part 4: Guidance and interpretation [211].
- IEC 62304 – Medical device software – Software life-cycle processes [212].
- ISO 14971 – Medical devices – Application of risk management to medical devices [213].
- IEC 62366 – Medical devices – Application of usability engineering to medical devices [214].

- ISO 13485 – Medical Devices – Quality Management. Requirements for regulatory purposes [215].
- IEC 81001-5-1 – Health software and health IT systems safety, effectiveness, and security – Part 5-1: Security – Activities in the product life cycle [216].
- IEC 62133-2 – Secondary cells and batteries containing alkaline or other non-acid electrolytes – Safety requirements for portable sealed secondary cells and batteries made from them for use in portable applications – Part 2: Lithium systems [217].
- IEC 60086-4 – Primary batteries---Part 4: Safety of lithium batteries [218].

Figure 3.1-1 shows the relationship between typical modules of an embedded medical device and the regulations that apply to them.

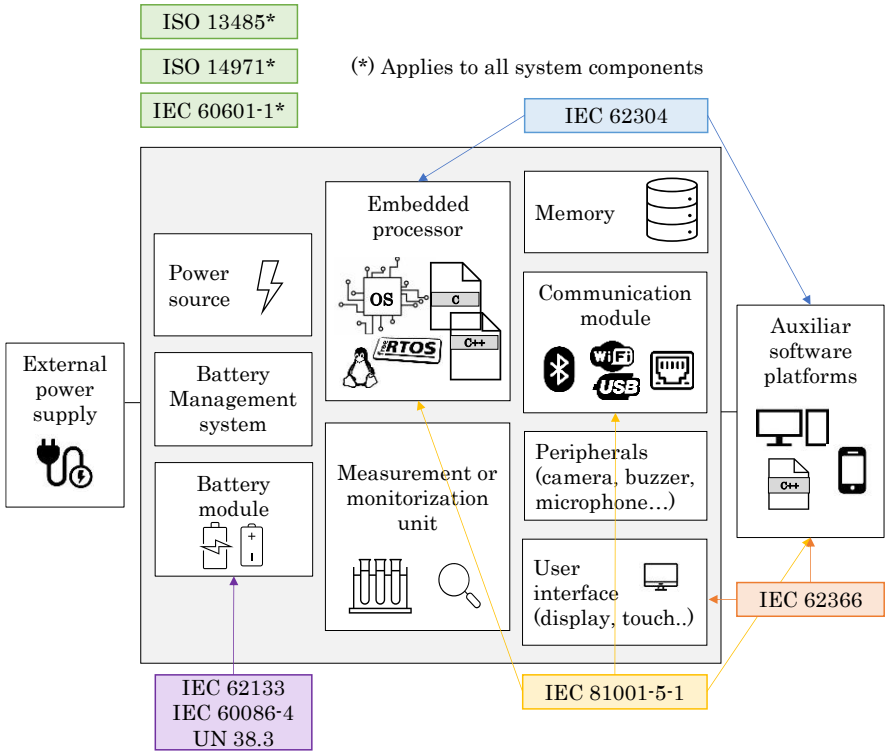


Figure 3.1-1: Embedded medical device modules and their regulations.

Although this thesis reviews the requirements associated with the design and development phases of medical devices for their subsequent commercialisation in the European market, adopting

the new MDR and IVDR requirements brings the European regulations closer to the needs of the FDA.

Most design and development requirements in the subsequent sections apply to the European and US markets. Both regulations recognise the main standards reviewed in this work, IEC 60601-1, IEC 62304, IEC 62366, IEC 81001-5-1 and ISO 14971. In addition, the FDA requires adopting a quality management system based on ISO 13485 [219].

The following sections review the standards identified as applicable to embedded systems. Table 3.1-1 lists the standards and amendments reviewed throughout this thesis.

Table 3.1-1: Standards and Amendments reviewed during this thesis.

Standard	Amendment
IEC 60601-1:2005	AMD1: 2012, AMD2: 2020
IEC 62304:2006	AMD1: 2015
ISO 14971:2019	AMD11: 2021
IEC 62366-1:2015	AMD1: 2020
ISO 13485:2016	AMD11: 2021
IEC 81001-5-1:2021	
IEC 62133-2:2017	AMD1: 2021
IEC 60086-4:2019	

3.2 Medical Device Regulation (MDR)

The 2017/745 [220] Medical Device Regulation (MDR) defines the rules concerning placing medical devices on the market for use in humans. This regulation covers medical devices, accessories for medical devices and clinical investigations related to medical devices.

This regulation defines a **medical device** as *any instrument, apparatus, appliance, software, implant, reagent, material or other article intended by the manufacturer to be used, alone or in combination, for human beings for one or more of the following specific medical purposes:*

- *Diagnosis, prevention, monitoring, prediction, prognosis, treatment or alleviation of disease.*
- *Diagnosis, monitoring, treatment, alleviation of, or compensation for, an injury or disability.*
- *Investigation, replacement or modification of the anatomy or of a physiological or pathological process or state.*
- *Providing information by means of in vitro examination of specimens derived from the human body, including organ, blood and tissue donations.*

and which does not achieve its principal intended action by pharmacological, immunological or metabolic means, in or on the human body, but which may be assisted in its function by such means.

The following products shall also be deemed to be medical devices:

- *Devices for the control or support of conception.*
- *Products specifically intended for the cleaning, disinfection or sterilisation of devices.*

3.2.1 Structure of the MDR

The MDR is structured into 10 chapters and 12 annexes, which are divided as follows:

- Part 1 - Scope and definitions:

This first chapter describes the purpose, field of application and explanations of the main terms.

- Part 2 - Making available on the market and putting into service of devices, obligations of economic operators, reprocessing, CE marking, and free movement:

This section defines the articles related to the commercialisation and placing into service of the products, distance sales, use of harmonised standards, general obligations of manufacturers and representatives, obligations of importers and distributors, CE conformity declaration and free circulation of the products.

- Part 3 - Identification and traceability of devices, registration of devices and of economic operators, summary of clinical safety performance, European database on medical devices:

This section defines articles on supply chain identification, medical device nomenclature, unique product identification system, product registration, electronic registration systems, manufacturers, authorised agents and importers, a summary of safety and clinical performance, etc.

- Part 4 - Notified bodies:

This section defines articles related to notified bodies, their responsible authorities, requirements, subsidiaries and subcontracting, the appointment of experts, linguistic requirements, designations and notifications, coordination and applicable fees, among others.

- Part 5 - Classification and conformity assessment:

Part 5 defines the classification of products and the procedures for conformity assessment, detailing aspects such as the involvement of notified bodies in this procedure, the mechanisms for the scrutiny of the conformity assessment, certificates of conformity, etc.

- Part 6 - Clinical evaluation and clinical investigations:

This section details articles related to clinical evaluation, general requirements concerning clinical studies, damage compensation, etc.

- Part 7 - Post-market surveillance, vigilance and market surveillance:

Regarding post-market monitoring, it defines the monitoring system, plan and report. It establishes the incident reporting and corrective actions, the incident analysis and the electronic surveillance system.

Finally, aspects such as evaluating products with suspected unacceptable risks, non-compliance with regulations and good administrative practices are discussed for market control.

- Part 8 - Cooperation between member states, medical device coordination group, expert laboratories, expert panels and device registers:

This chapter aims to define aspects related to competent authorities, cooperation, coordination groups, provision of scientific, technical and clinical advice, conflict of interest, etc.

- Part 9 - confidentiality, data protection, funding and penalties:

Confidentiality, data protection, charging fees or financing activities related to the designation and supervision of notified bodies are some of the articles defined in this section.

- Part 10 - final provisions:

This section defines articles related to committee procedures, the exercise of delegation, amendment of other directives and regulations, repeals and the definition of entry into force and date of application.

- Annexes:

The annexes cover general safety and performance requirements, technical documentation, EU Declaration of Conformity, CE conformity marking, product classification rules, clinical evaluations and investigations, among others.

3.2.2 Medical device classification according to MDR

Medical devices can be classified into different categories based on their inherent risks. The manufacturer decides the classification and may consult the notified body in case of doubt. In Europe, the risk is defined based on the human body's vulnerability to the device, intended purpose, and use duration. However, each regulatory authority has a different classification. According to the MDR, medical devices are classified into four categories.

Table 3.2-1: Medical Device classification according to MDR.

Class	Risk level	Device type	Examples
I	Low	Non-invasive, sterile and for measurement.	Insoles for feet, adhesive plasters, etc.
IIa	Medium	Short-term invasive devices with no significant effect on organisms and fluids.	Scalpel, ultrasound machine, feeding probe, etc.
IIb	Medium – High	Long-term invasive devices with a significant effect on organisms and fluids.	Intraocular lenses, external defibrillator, dialyser, X-ray, etc.
III	High	It can be potentially life-threatening, fully absorbed in the human body, implantable, with a medicine of animal source.	Breast implants, implantable pacemakers, cochlear implants, defibrillators, etc.

Annexe VIII of the MDR clarifies the classification of devices, as sometimes, this task is difficult. This annexe details the difference between invasive and non-invasive devices, the definition of short and long use time, etc. It also provides specific details on classifying medical devices that only contain software.

Regarding medical devices comprised only of software, it is defined that software intended for therapeutic or diagnostic decisions is classified as IIa, unless it may cause death or irreversible deterioration of the state of health, in which case it is considered class III. If it may cause severe damage to the state of health or surgical intervention, it is categorised as IIb.

3.2.3 Pre-commercialization requirements

- Clinical Evaluation:

The MDR defines **clinical evaluation** as *a systematic and planned process to continuously generate, collect, analyse and assess the clinical data pertaining to a device in order to verify the safety and performance, including clinical benefits, of the device when used as intended by the manufacturer*.

This evaluation shall be the confirmation of conformity with the general safety and performance requirements under normal conditions of product use. This evaluation shall include assessing undesirable effects and the acceptability of the risk and benefit balance. The clinical evaluation shall be based on clinical data provided together with the specification and justification of the manufacturer concerning the level of performed clinical tests.

Clinical evaluation is required for all devices unless the manufacturer justifies it in its technical documentation. This can be done based on risk management, the interaction between the product and the human body, the intended clinical performance and the manufacturer's statements.

It is also mandatory to have the clinical evaluation report updated throughout the product's life cycle. This update must be performed using data from the post-market clinical follow-up plan.

- Clinical Investigation:

The MDR defines **clinical investigation** as *any systematic investigation involving one or more human subjects, undertaken to assess the safety or performance of a device*.

Clinical investigation reinforces the rules governing clinical evaluation throughout the lifetime of medical devices to ensure the safety of patients and consumers. This evaluation is not always mandatory. It is only necessary if additional data in the clinical evaluation is needed to ensure the device is safe and effective.

- Notified bodies:

According to the MDR, **notified bodies** are defined as *a conformity assessment body designated in accordance with this Regulation*.

The notified bodies verify that the product satisfies the general safety and performance requirements. They are entities that are independent of the authorities that assess the products. The competent authorities of the member states designate these notified bodies. The medical device manufacturer is free to choose the notified body responsible for the conformity assessment of the product.

- Conformity assessment:

The MDR defines **conformity assessment** as *the process demonstrating whether the requirements of this Regulation relating to a device have been fulfilled*.

Manufacturers must perform a conformity assessment before placing the device on the market. A conformity assessment body, such as a notified body, can complete this process. The conformity assessment will be different depending on the category of the medical device:

Class I devices do not require to be reviewed by a notified body (except for sterile products with a measuring function or reusable surgical instruments). They require a quality system which does not need to be certified by a notified body.

The conformity assessment of these class IIa devices is based on a technical documentation review, at least for one representative product per product category. These devices require a quality system audited by a notified body.

Regarding class IIb, technical documentation review is required for at least one representative product per generic product group or product. A quality system audited by a notified body is also required.

Finally, for class III, a review of each product's technical documentation and the quality system's audit by a notified body is required.

3.2.4 Post-commercialization requirements

Manufacturers must comply with different surveillance and vigilance requirements after placing their devices on the market. The most relevant ones are listed below:

- Documentation maintenance:

According to the MDR, it is required to have documentation (technical, CE certificate...) available for the competent authority.

- Product complaints or incident reports:

Suppose any complaints or reports of suspected incidents related to the product are received from healthcare professionals, patients or users. This information shall be immediately transmitted to the manufacturer, its authorised representative and the importer. Maintaining a register of complaints, non-compliant products, and recalls will also be required.

- Non-compliances:

Suppose an importer or distributor detects that the product is not compliant with the regulation. In that case, it is necessary to inform the manufacturer, authorised representative and importer and cooperate with them to take the required corrective actions. The competent authorities must be notified when a product presents a severe risk.

- Serious incidents:

The device manufacturer shall inform the relevant authorities of any serious incidents associated with devices placed on the European Union market. This statement shall not apply to side effects clearly documented in the product information, quantified in the technical documentation and included in the trend report.

Moreover, any corrective safety action regarding products placed on the Union market should also be reported, including those carried out in third countries when they concern products also marketed in the European Union.

- Collaboration with the competent authority:

The manufacturer, distributors and importers have to provide the competent authority with the requested information as well as free samples of the product when necessary.

- Post-commercialisation monitoring:

The manufacturer must implement and keep up to date a post-market surveillance system recording all performed activities. Depending on the device class, the MDR defines different scopes to be covered by the post-market report. The Post-Market Surveillance

Report (PMSR) and the Post-market Safety Update Report (PSUR) are two main reports.

The PMSR includes results and conclusions from post-market surveillance data and the post-market plan. It also defines the preventive and corrective actions to be taken. PSUR collects the results and findings derived from vigilance, post-market surveillance, and Post-Market Clinical Follow up (PMCF). Depending on the device's classification, different post-market documentation will be required.

3.2.5 Traceability and registration requirements

Regarding traceability and registration, two essential concepts are identified and described below.

- Unique Device Identification system (UDI system):

The European regulation defines a unique identification system called UDI. This identification system aims to ensure the effective traceability of medical devices in the European Union. It also seeks to increase the effectiveness of post-market safety activities, reduce medical errors and product counterfeiting, and improve purchasing and waste disposal policies. Finally, it aims to increase stock management efficiency for healthcare institutions and other economic operators.

The manufacturer assigns this indicator to all products before placing them on the market. The UDI must appear on the product label and all upper levels of the packaging. All economic agents must store and retain the UDI of purchased or supplied products.

The UDI consists of two components Device Identification (UDI-DI) and Product identification (UDI-PI).

The UDI-DI is a product identifier specific to a manufacturer and a product. This indicator comprises a prefix of the issuing entity, manufacturer identification code, product code and a control digit.

The UDI-PI is the production identifier, identifying a single production unit.

- European database on medical devices (Eudamed):

In 2023, Eudamed, the European database on medical devices, will be launched. This platform will aim to increase transparency, improve access to information for the public and healthcare professionals and strengthen coordination between member states. The UDI of the devices must be registered on this platform.

3.2.6 MDR Requirements for Medical Device Design and Development

The MDR covers all phases of the life cycle of a medical device [208]. This thesis aims to extract only the requirements associated with the design and development phases. To fulfil the needs of these phases, the MDR refers to standards discussed in the following sections, such as ISO 13485, IEC 60601 or IEC 62304 [221].

As a starting point, based on this regulation, it will be necessary to identify the class of the device as the scope of the different standards varies. Identifying all the specific rules that apply to the device will also be necessary.

Regul. Des. Req. 1 Identify the classification of the new medical and applicable regulations.
Regul. Des. Req. 2 Perform Clinical Evaluation and Clinical Investigation (if required).

3.3 In-Vitro Medical Device Regulation (IVDR)

The 2017/746 [3], In Vitro Medical Device Regulation (IVDR) defines the rules for placing on the market in-vitro medical devices.

According to the IVDR, the **in-vitro health product** is defined as *any medical device which is a reagent, reagent product, calibrator, control material, kit, instrument, apparatus, piece of equipment, software or system, whether used alone or in combination, intended by the manufacturer to be used in vitro for the examination of specimens, including blood and tissue donations, derived from the human body, solely or principally for the purpose of providing information on one or more of the following:*

- *Concerning a physiological or pathological process or state.*
- *Concerning congenital physical or mental impairments.*
- *Concerning the predisposition to a medical condition or a disease.*
- *To determine the safety and compatibility with potential recipients*
- *To predict treatment response or reactions.*
- *To define or monitoring therapeutic measures.*

In addition, all in-vitro products containing a medical device are regulated by the MDR. Specifically, IVDR requirements apply only to the product part that constitutes an in-vitro product.

3.3.1 Structure of the IVDR

IVDR is structured in 10 chapters providing equivalent information to that presented in the MDR. In this case, with a specific focus on the IVDR. It also has 15 additional annexes.

- Part 1 - Introductory provisions.
- Part 2 - Making available on the market and putting into service of devices, obligations of economic operators, CE marking, and free movement.
- Part 3 - Identification and traceability of devices, registration of devices and economic operators, summary of safety and clinical performance, European database on medical devices.
- Part 4 - Notified bodies.
- Part 5 - Classification and conformity assessment.
- Part 6 - Clinical evidence, performance evaluation and performance studies.
- Part 7 - Post-market surveillance, vigilance, and market surveillance.
- Part 8 - Cooperation between member states, medical device coordination group, EU reference laboratories and device registers.
- Part 9 - Confidentiality, data protection, funding, and penalties.
- Part 10 - Final provisions.
- Annexes.

3.3.2 In-Vitro Medical device classification

In-vitro devices can be classified into four categories depending on the intended purpose of use and the risk they present. To correctly categorise in-vitro devices, Annex VIII defines seven rules.

Table 3.3-1: In-vitro medical device classification according to IVDR.

Class	Individual Risk level	Public health risk	Examples
A	Low	Low	Specimen receptacles, laboratory instruments, buffer solutions, etc.
B	Medium	Low	Blood chemistry, pregnancy tests, etc.
C	High	Medium	Oncological markers, dangerous infectious diseases, etc.
D	High	High	Blood safety, high-risk infectious diseases, etc.

3.3.3 Pre-commercialization requirements

The main difference in pre-market requirements between the MDR and the IVDR is that MDR, as pre-market data, requires a clinical evaluation report based on clinical evidence or clinical investigation. However, the IVDR requires performance evaluation and performance studies. Furthermore, the conformity assessment will differ depending on the device's classification.

- Performance evaluation and performance studies:

According to the IVDR, performance evaluation must be conducted to commercialize an in-vitro device. These tests confirm compliance with general safety and performance requirements under the standard conditions of the device's intended use.

The assessment of the performance shall follow the defined methodology to demonstrate scientific effectiveness, analytical performance, and clinical performance. Clinical performance studies must also be conducted unless their performance can be justified using clinical performance data from other sources.

The obtained data, their evaluation and the clinical evidence, should be documented in the performance evaluation report. The

performance evaluation and documentation must be updated throughout the product's life cycle. This documentation must be updated with the data obtained from the execution of the post-market clinical follow-up plan.

The performance evaluation report for Class C and D devices shall be updated whenever necessary, but at least annually. The safety and clinical performance summary shall be updated as soon as possible, when necessary.

- Conformity assessment:

The conformity assessment, as in MDR, will be different depending on the classification of the medical device:

Class A devices require preparing all technical documentation and self-declare conformity with the IVDR. This device does not require an audit of a notified body.

Class B requires Quality Management System Assurance and the assessment of technical documentation per category device. It also requires an audit of a notified body.

Regarding class C, there are two options for obtaining the conformity assessment. On the one hand, these devices require Quality Management System (QMS) assurance and the review of technical documentation per generic device. An audit by a notified body is also required. On the other hand, it is possible to get the conformity assessment by making a technical documentation review and physical or laboratory tests on a representative sample of the generic device group. Additionally, the assurance of the manufacturing process quality is required, including a quality system audit of the production of the selected device.

Finally, for class D, there are also two options. The first is through assurance of the QMS and the assessment of technical documentation for each device. All of this needs to be audited by a notified body. The second is by examination, for that is necessary to review the technical documentation, physical or laboratory tests on the devices and the assurance of the production quality, which involves a quality systems audit of the production devices.

3.3.4 Post-commercialization requirements

In MDR, a continuous post-market clinical follow-up is necessary, and IVDR requires post-market surveillance and vigilance. As in MDR, the IVDR requires performing a post-market surveillance report, periodic safety update report, post-market follow-up and vigilance reporting, depending on the device's classification.

3.3.5 Traceability and registration requirements

As defined by the MDR, one of the objectives of the new IVDR regulation is to implement a European database for medical devices (EUDAMED). This database will assign a product identifier (UDI) to be digitally recorded.

3.3.6 IVDR Requirements for In Vitro Medical Device Design and Development

This regulation is equivalent to the MDR but for in vitro devices. As with the MDR, this regulation refers to different standards regulating design and development phases. In this case, it will also be necessary to identify the classification of the device and all the specific regulations that apply to it.

Regul. Des. Req. 3 Identify the classification of the new medical and applicable regulations.
Regul. Des. Req. 4 Complete Performance Evaluation and Performance Studied (if required).

3.4 IEC 60601 - Basic Safety and Essential Performance

IEC 60601 is a series of technical standards intended to ensure medical electrical devices' safety and good performance. These series are published by the International Electrotechnical Commission (IEC) and include a primary or general standard (IEC 60601-1), about ten collateral standards (IEC 60601-1-x) specifying general requirements for some devices, and about 80 particular standards (IEC 60601-2-x) specifying particular requirements for some medical products. The specifications defined by the particular

standards prevail over those described in the general and collateral standards. In addition, technical reports (IEC 60601-4-x) are also published, and they serve as guidance for different aspects of medical devices. Table 3.4-1 presents the standards included in the IEC 60601 family.

Table 3.4-1: IEC 60601 standard series

Primary standards		
IEC 60601-1 General Safety		
Collateral standards		
IEC 60601-1-2 Electromagnetic disturbances	IEC 60601-1-3 Radiation protection in X-ray equipment	IEC 60601-1-6 Usability
IEC 60601-1-8 Alarm Systems	IEC 60601-1-9 Environmentally conscious design	IEC 60601-1-10 Physiological Closed-loop controllers
IEC 60601-1-11 Home Healthcare Environment	IEC 60601-1-12 Emergency Medical Services	
Particular standards		
IEC 60601-2-1 Electron accelerators	IEC 60601-2-2 High-frequency equipment	IEC 60601-2-3 Short-wave therapy equipment
IEC 60601-2-4 Cardiac defibrillators	IEC 60601-2-5 Ultrasonic physiotherapy	IEC 60601-2-6 Microwave therapy equipment
IEC 60601-2-8 Therapeutic X-ray equipment	IEC 60601-2-10 Nerve and muscle stimulators	IEC 60601-2-11 Gamma beam therapy equipment
IEC 60601-2-12 Critical care ventilators	IEC 60601-2-13 Anaesthetic systems	IEC 60601-2-16 Haemodialysis, diafiltration, and haemofiltration
IEC 60601-2-17 Automatic brachytherapy	IEC 60601-2-18 Endoscopic equipment	IEC 60601-2-19 Infant incubators
IEC 60601-2-20 Infant transport incubators	IEC 60601-2-21 Infant radiant warmers	IEC 60601-2-22 Surgical, cosmetic, therapeutic and diagnostic laser
IEC 60601-2-23 Transcutaneous partial pressure monitoring	IEC 60601-2-24 Infusion pumps and controllers	IEC 60601-2-25 Electro-cardiographs

IEC 60601-2-26 Electro-encephalography	IEC 60601-2-27 Electro-cardiographic	IEC 60601-2-28 X-ray tube assemblies
IEC 60601-2-29 Radiotherapy simulators	IEC 60601-2-31 External pacemakers	IEC 60601-2-33 Magnetic resonance equipment
IEC 60601-2-34 Invasive blood pressure monitoring	IEC 60601-2-36 Extracorporeally induced lithotripsy	IEC 60601-2-37 Ultrasonic medical diagnostic and monitoring
IEC 60601-2-39 Peritoneal dialysis equipment	IEC 60601-2-40 Electro-myographs and evoked response equipment	IEC 60601-2-41 Surgical luminaires and luminaires for diagnosis
IEC 60601-2-43 X-ray equipment for interventional procedures	IEC 60601-2-44 X-ray equipment for computed tomography	IEC 60601-2-45 Mammographic x-ray and stereotactic devices
IEC 60601-2-46 Operating tables	IEC 60601-2-47 Ambulatory electro-cardiography	IEC 60601-2-49 Multifunction patient monitoring
IEC 60601-2-50 Infant phototherapy equipment	IEC 60601-2-52 Medical beds	IEC 60601-2-54 X-ray equipment for radiography and radioscopy
IEC 60601-2-57 Non-laser light source equipment	IEC 60601-2-62 High-intensity therapeutic ultrasound (HITU)	IEC 60601-2-63 Dental extra-oral X-ray equipment
IEC 60601-2-64 Light ion beam medical electrical equipment	IEC 60601-2-65 Dental intra-oral X-ray equipment	IEC 60601-2-66 Hearing instruments and systems
IEC 60601-2-68 X-ray-based image-guided radiotherapy	IEC 60601-2-75 Photodynamic therapy	IEC 60601-2-76 Low energy ionized gas haemostasis
IEC 60601-2-83 Home light therapy	IEC 60601-2-84 Emergency and transport ventilators	
Technical reports		
IEC 60601-4-1	IEC 60601-4-2 Electromagnetic immunity	IEC 60601-4-3 Unaddressed safety aspects

MEE and MES employing a degree of autonomy		
IEC 60601-4-4 Guidance for writers of particular standards related to alarm systems	IEC 60601-4-5 Safety-related technical security specifications	

The general standard, the IEC 60601-1 [222], defines the general requirements for basic safety and essential performance. This standard appeared in 1977 with its first edition (Ed. 1). In 1988, a second version (Ed. 2) was generated, which focused on ensuring safety in the vicinity of the patient. The third edition (Ed. 3) was published in 2005 and is still in force. In this edition, protection has been extended to patients and equipment operators. In addition, a major review of the standard was carried out in 2012, which clarified several ambiguities generated by the evolution of technology. A new standard edition is expected to be published in 2024 [210]. Table 3.4-2 lists the different versions and amendments of the IEC 60601-1 standard.

Table 3.4-2: Current IEC 60601-1 standard editions and amendment.

Standard	Amendment
IEC 60601-1:1977 Ed. 1	AMD1:1984
IEC 60601-1:1988 Ed. 2	AMD1:1991 AMD2:1995
IEC 60601-1:2005 Ed. 3	AMD1:2012 AMD2:2020

Different editions of the standards are widely accepted in many countries, all of Europe (Ed. 3.1), Canada (Ed. 3.1), the USA (Ed 3.1), Japan (Ed. 3.1), China (Ed. 2), Brazil (Ed. 3.1), South Korea (Ed. 3.1) and Taiwan (Ed. 2) recognise the IEC 60601 standards [222].

3.4.1 Structure of the IEC 60601 series standard

The content of all IEC 60601 family standards is similar. The five most important parts are listed below:

- Part 1 - Scope, object and related standards: This chapter describes the standard's scope, the objective, and the associated standards.
- Part 2 – Normative references: It usually indicates the references of the standards to which it refers. It also specifies the amendment of each standard referred to.
- Part 3 – Terminology and definition: It identifies the main terms referenced in the standard.
- Part 4 – Requirements: It is divided into several sections, and all the specific requirements of the standard are defined.
- Part 5 – Annexes: It includes informative or normative information to be added to what is defined in the section on general requirements.

3.4.2 Scope of IEC 60601-1

The IEC 60601-1 standard generally applies to medical electrical devices connected to or in physical contact with the patient voluntarily or involuntarily. For defining the application of the standard, two types of devices are defined, Medical Electrical Equipment (MEE) and Medical Electrical Systems (MES):

- Medical electrical equipment (MEE):

According to the definition of the standard, a **MEE** is *an electrical equipment having and applied part or transferring energy to or from the patient or detecting such energy transfer to or from the patient and which is: (a) provided with not more than one connection to a particular supply mains; and (b) intended by its manufacturer to used in the diagnosis, treatment or monitoring of a patient; or for compensation or alleviation of disease, injury or disability.*

Therefore, if a device is electrically connected to a patient, it is considered MEE, for example, an electrocardiogram. An electrical device that does not require direct electrical contact with the patient's body but must be in touch with the patient during the test is also considered MEE.

- Medical electrical systems (MES):

IEC 60601-1 defines a **MES** as a *combination, as specified by its manufacturer of items of equipment, at least one of which is MEE*

to be interconnected by functional connection or by use of a multiple socket-outlet.

If the device has more than one electrical connection, it is considered an MES, not an MEE. On the other hand, it does not cover the following categories:

- In vitro diagnostic equipment is covered by the IEC 61010.
- Implantable parts of active implantable medical devices, covered by ISO 14708.
- Medical gas pipeline systems are covered by ISO 7396-1.

However, it is necessary to identify which regulations are applicable, as the relevant rules for a particular device are often a combination of several.

3.4.3 Device classification

IEC 60601-1 classifies devices depending on the type and degree of protection against electric shocks. Regarding the type of protection, there are two types of devices. Depending on whether they are powered externally or internally. At the same time, externally powered devices are divided into two categories:

- Class I: These are devices protected against electric shock by additional protection over the essential insulation. That is the earth connection of exposed conductive parts.
- Class II: The devices are protected against electric shock by double or reinforced insulation without an earth connection.

According to the degree of protection of the device against electric shock, medical devices can be divided into two categories. This categorisation only applies to devices considered as applied parts, i.e. devices designed to be in physical contact with the patient.

- Type F: It is an electrically earth-isolated device. There are two different categories BF and CF.
    - Type BF: These devices have conductive contact with the patient but not directly with their heart. They require less sophisticated protection than CF-type devices.

    - Type CF: These devices have the most advanced protections, usually in direct conductive contact with the heart.
- Type B: They are devices that are referenced to the earth. They are not in conductive contact with the patient and therefore have less stringent protections.

3.4.4 General requirements of IEC 60601-1

IEC 60601-1 aims to achieve basic safety and essential performance. This standard defines **basic safety** as *freedom from unacceptable risk directly caused by physical hazards when MEE is used under normal condition and single fault condition* and essential performance as *performance of a clinical function, other than that related to basic safety, where loss or degradation beyond the limits specified by the manufacturer results in an unacceptable risk*.

To guarantee basic safety and essential performance for the patient, operators and device environment, IEC 60601-1 defines the safety requirements to be met by electromedical devices.

These standards are based on the ISO 14971 standard, which defines the risk management of medical devices. IEC 60601 identifies electrical hazards that may exist in devices, so the different standards of the family set out technical requirements to be met, verification procedures and conditions, as well as acceptance criteria. After a risk analysis of the device, it is possible to identify electric safety risks not contemplated in this standard or cases where the risk may exist but not the technical requirement to be fulfilled. In these cases, the device manufacturer will perform the risk analysis and establish the necessary measures to reduce the identified risk.

This standard also defines some critical aspects of the design, such as risk management, the expected service life, the device fault conditions and the required protections against different sources of hazards (electrical, mechanical, radiation, temperature, etc.). For testing, it defines test requirements related to environmental conditions (temperature, humidity, atmospheric pressure) or test-related supply voltage.

- General requirements for testing:

The standard identifies that testing should be performed in the least favourable use case of operation. This scenario should be determined by performing a risk analysis.

- Equipment identification, marking and documents:

IEC 60601-1 defines the identification of equipment and documentation. It describes the requirements for measurement units, symbols, durability and legibility of the instructions for use. Additionally, it defines how the marking of the device should be.

- Protection against electrical hazards:

Protection against electrical hazards is another defined point; the standard specifies the values for different working **voltages of creepage distance** as *the shortest distance along the surface of the insulating material between two conductive parts* and air clearance, defined as the *shortest path in air between two conductive parts*.

- Protection against mechanical hazards:

An MEE's risk is not only electrical; it can also be mechanical. Therefore, the standard covers moving parts, trapping zones, emergency stops and stability requirements. The standard also defines different tests to be carried out, such as the device falling to the ground under certain conditions.

- Protection against radiation hazards:

The standard covers protection against Infra-Red (IR), Ultra-Violet, Microwave, X-rays and Particle Radiation. In addition, since several devices intentionally emit radiation, some particular standards define their requirements and tests.

- Protection against excessive temperature and other hazards:

The standard defines maximum temperature requirements; these vary depending on the material, the period they contact the patient or operator, etc. Likewise, fire protection, liquid overflow or biocompatibility requirements are also stated. More specifically, conditions related to biocompatibility are assessed with a set of

standards under ISO 10993 (Biological evaluation of medical devices).

- Accuracy of controls:

This section introduces the concept of usability in the medical device engineering process. It identifies the key to minimising use errors and their associated risks.

- Hazardous situations and fault conditions:

The standard identifies risks such as the emission of lames, poisonous or ignitable substances not permitted under fault conditions. The fact that this list does not cover a risk does not mean risk analysis and mitigation actions are unnecessary.

- Programmable Electrical Medical Systems (PEMS):

This standard identifies safety requirements for devices containing firmware or software. It also cites as a reference standard the IEC 62304, which details the software lifecycle.

- ME Systems:

It also identifies safety requirements to be met by the equipment when several ME devices are connected with power or data cables. For these cases, it also specifies the tests to be carried out.

- Electromagnetic Compatibility (EMC):

The standard covers how performance and safety can be affected by electrostatic discharge (ESD), radiated and conducted immunity, and power line disturbances. The requirements to be fulfilled are listed in the collateral standard IEC 60601-1-2.

3.4.5 Safety Requirements for Medical Device Design and Development

First, it will be necessary to identify the device's classification that depends on the type and degree of protection against electric shocks.

It is also required to identify which collateral and particular standards apply to the device to extract specific technical requirements such as the device's performance regarding

electrostatic discharge (ESD), radiated and conducted immunity or during drop tests.

Once the requirements have been identified, the standard defines that it will be necessary to perform tests to ensure the functional safety of the device.

Regul. Des. Req. 5 Identify device safety classification.
Regul. Des. Req. 6 Identify technical requirements to be met (defined in IEC 60601 family).
Regul. Des. Req. 7 Perform safety tests to guarantee functional safety.
Regul. Des. Req. 8 Perform device safety risk analysis according to ISO 14971 (see Figure 3.6-1).

3.5 IEC 62304 - Medical Device Software

IEC 62304 standard [212] is published by the International Electrotechnical Commission (IEC). It provides processes to develop software for medical devices. It defines lifecycle processes with activities and tasks necessary to safely design, develop, and maintain medical device software. This standard references ISO 14971 on medical device risk management; both are aligned.

The first version of this standard was published in 2006. Later an amendment was generated in 2012, known as Edition 1.1. Currently, edition 2 of the standard is being drafted and is planned to be released in 2023; it is expected that this version will also include in its processes software equipment that is not considered to be medical devices [223].

3.5.1 Structure of the IEC 62304 standard

The IEC 62304 standard consists of 5 parts. The contents of these are summarised below:

- Part 1 – Scope: This chapter presents the purpose, the field of application and the associated standards.
- Part 2 – Normative references: The referenced standards are identified by defining the specific version of the standard.

- Part 3 – Terms and definitions: The principal terms used in the standard are defined.
- Part 4 – Requirements, processes and tasks: The requirements related to the quality management system, risks and safety classification of software, as well as the design, maintenance, risk management and troubleshooting processes associated with the software of a medical device, are detailed.
- Part 5 – Annexes: Further information is added to that detailed in the standard.

3.5.2 Scope of IEC 62304

IEC 62304 is applied to all medical devices in which software is available as a core or relevant part of the medical device. It also applies when the device is composed entirely of software components.

This standard covers the software development and maintenance phases. However, it does not cover the final validation and dissemination of the device.

3.5.3 Software safety classification

The manufacturer must classify each software system according to its safety class (A, B or C). This classification depends on the possible effects on the patient or other nearby people.

- Class A: Software that cannot cause injury or harm to health.
- Class B: The software can cause injury, but not severe.
- Class C: It can cause death, and severe injury is possible.

In case the use of hardware or other mechanisms can mitigate or reduce the risk of injury, the class of the software can be reduced. In addition, the risk analysis performed according to ISO 14971 must include each software system's classification.

3.5.4 Requirements, processes and tasks

The software of a medical device can be a risk source for the patient or the operator. Risk scenarios can especially arise during the product's development and maintenance phases. Therefore, this

standard includes processes and tasks to minimise risk throughout the lifecycle of medical device software.

Also, as a general requirement, the standard identifies the need for a QMS. ISO 13485 is recognised as a reference quality management system that can demonstrate the ability to design, develop and manufacture a medical device. Furthermore, it identifies that the product manufacturer must manage risk according to the ISO 14971 standard.

The main processes involved in this standard are Software Development, Software Maintenance, Software Risk Management, Software Configuration Management and Software Problem Resolution. In the following subsections, these processes will be detailed.

3.5.4.1 Software development process

The software development process consists of 8 processes and several sub-processes that must be carried out during the development phase. Not all of them are required, as their implementation also depends on the software classification to be developed. Figure 3.5-1 shows the processes and tasks IEC 62304 defines for the product development phase.

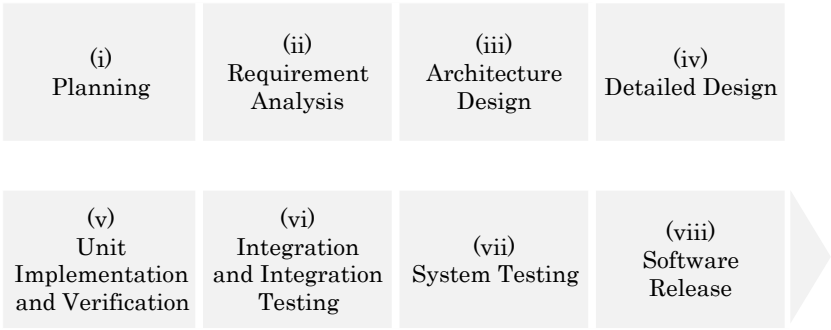


Figure 3.5-1: Software development processes and activities.

The development process begins with *(i) Software Development Planning*. The manufacturer must define a development plan considering processes to be used during the development phase, the deliverables to be generated, the traceability between requirements and control measures, how configuration management and software changes will be carried out, and the resolution of problems arising

during the development phase. The development plan must also reference the standards, methods and tools on which the development will be based. It will also be necessary to define the software verification and integration planning. The development plan must be updated during the software product development.

Once the development plan has been defined, the *(ii) Software Requirements Analysis* must be carried out. Software requirements must be defined and documented for this, detailing functional and non-functional requirements, system inputs and outputs, interfaces, alarms, warnings, security requirements, etc. The system requirements must be verified, re-evaluated and updated throughout the software development life cycle.

This process is followed by *(iii) Architectural Software Design*, which involves turning software requirements into a documented architecture describing the software's structure, elements, interfaces and SOUPs. **SOUP** is defined as a *software item that is already developed and generally available and that has not been developed for the purpose of being incorporated into the product*. Therefore, these elements often do not have adequate documentation of the development process.

The architecture design is followed by *(iv) Detailed Software Design*. The manufacturer must improve the software architecture by representing it in software units and blocks that cannot be divided into other elements.

The next process is the *(v) Software Unit Implementation and Verification*. The implementation must be verified by defining the verification tests and the associated acceptance criteria.

Once implemented, carrying out (*vi) Software Integration and Integration Testing* are necessary. The manufacturer must perform integration and verification of the software according to the integration plan. The verification process must ensure that the integrated elements work as intended. The verification records must contain enough information to reproduce the test. It is also necessary to keep records of who has performed the tests and what the test results were.

The verification process continues with *(vii) Software System Testing*. It is necessary to verify the integrated software considering the different inputs and outputs of the system. The results of these tests must be recorded.

Finally, the *(viii) Software Release* is done. Once the verification process has been completed, the software can be released; this can be done after recording and evaluating the known residual faults. The generated software release must be documented.

Table 3.5-1 presents the requirements that each class of medical device software must meet.

Table 3.5-1: Software development process requirements.

Clause	Subclause	Class		
		A	B	C
(i) Software Development Planning	- Software development plan. - Keep the software development plan updated. - Software development plan reference to system design and development. - Software verification planning. - Software risk management planning. - Documentation planning. - Software configuration management planning.	X	X	X
	- Software integration and integration testing planning. - Supporting items to be controlled. - Software configuration item control before verification. - Identification and avoidance of common software defects.		X	X
	- Software development standards, methods and tools planning.			X
(ii) Software Requirement Analysis	- Define and document software requirements from system requirements. - Software requirements content. - Re-evaluate product risk analysis. - Update requirements. - Verify software requirements.	X	X	X
	- Include risk control measures in software requirements.		X	X

Clause	Subclause	Class		
		A	B	C
(iii) Software Architectural Design	- Transform software requirements into an architecture. - Develop an architecture for the interfaces of software items. - Specify functional and performance requirements of soup items. - Specify systems hardware and software required by soup item. - Verify software architecture.		X	X
	- Identify segregation necessary for risk control.			X
(iv) Detailed Software Design	- Subdivide software into software units.		X	X
	- Develop detailed designs for each software unit. - Develop detailed designs for interfaces. - Verify detailed design.			X
(v) Software Unit Implementation and Verification	- Implement each software unit.	X	X	X
	- Establish software unit verification process. - Software unit acceptance criteria. - Software unit verification.		X	X
	- Additional software unit acceptance criteria.			X
(vi) Software Integration and Integration Testing	- Integrate software units. - Verify software integration. - Software integration testing. - Software integration testing content. - Evaluate software integration test procedures. - Conduct regression tests. - Integration test record contests. - Use software problem-resolution process.		X	X
(vii) Software System Testing	- Establish tests for software requirements. - Use software problem-resolution process. - Retest after changes. - Evaluate software system testing. - Record software system test results.	X	X	X
	- Ensure software verification is complete.	X	X	X

Clause	Subclause	Class		
		A	B	C
(viii) Software Release	- Document known residual anomalies. - Document released versions. - Archive software. - Assure reliable delivery of released software.			
	- Evaluate known residual anomalies. - Document how the released software was created. - Ensure activities and tasks are complete.		X	X

3.5.4.2 Software maintenance process

The software maintenance process has three processes and must be performed regardless of the classification of the software.

The first process is the *(i) Establishment of Software Maintenance Plan*. The manufacturer must define the software maintenance plan. Procedures for device reception, documentation, evaluation, resolution of returns, and improvement of the developed software must be generated.

(ii) Problem and Modification Analysis task is also necessary. This phase requires evaluating and documenting the received feedback, analysing the change request and performing its approval. Once modifications have been implemented, users and regulatory bodies need to be informed.

The *(iii) Modification Implementation* task is based on the software development process. The eight processes defined for software development must be executed.

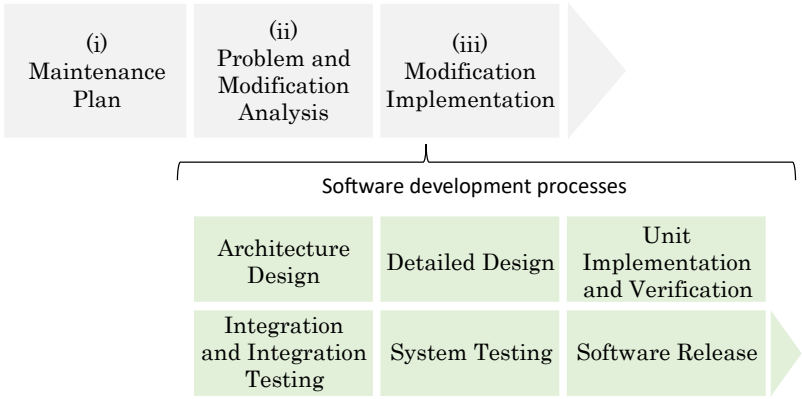


Figure 3.5-2: Software maintenance processes and activities.

3.5.4.3 Software risk management process

Software risk management processes are divided into four main processes. These aim to minimise the risks of medical device software.

Firstly, the *(i) Analysis of Software Contributing to Hazardous Situations* must be performed. The manufacturer shall identify the software elements that could contribute to a hazardous situation, identify the potential causes of failure, evaluate the public issue lists of all SOUPs and document the possible sources and the sequences of events that could lead to a hazardous situation.

Following the analysis, *(ii) Risk Control Measures must be defined*. For each potential cause of risk, control measures must be defined, documented and implemented.

After implementation, the *(iii) Verification of Risk Control Measures* needs to be performed. The software implementation must be verified and documented. The manufacturer must also register and maintain the traceability of the software hazards.

Finally, *(iv) Risk Management of Software Changes* is done. The manufacturer must analyse the software changes and evaluate their safety. It is also required to analyse how the modifications may impact the existing risk control measures.

The tasks of the risk management process have to be executed only on class B and C software devices, except for the safety analysis task, which is also required on class A devices.

3.5.4.4 Software configuration management process

The software configuration management process involves three tasks that must be carried out for all medical devices, regardless of the classification of their software.

First, *(i) Configuration Identification* is undertaken. The manufacturer has to identify configuration items (software units and SOUPs) and their software versions.

Then, *(ii) Change Control* must be performed. For this, change requests must be approved, implemented, and verified.

Finally, *(iii) Configuration Status Accounting* needs to be generated. The manufacturer must keep and maintain records that control the evolution of the system configuration.

3.5.4.5 Software problem resolution process

The problem-resolution process is based on eight tasks that must be carried out for all medical devices with software.

For each identified problem, the manufacturer must prepare *(i) Problem Reports* identifying their type, field of application and severity.

After generating the report, *(ii) Investigate the Problem*. When possible, the problem source must be identified, its relevance assessed and documented, and the change request created.

Then, it is necessary to *(iii) Advise Relevant Parties* and *(iv) Use a Change Control* process to approve and implement all change requests.

It is also essential to *(v) Maintain Records* that include problem reports, revisions, and verifications.

The manufacturer must also *(vi) Analyse Problems for Trends* to identify if there are significant patterns in the problem reports.

Another process to be executed is the *(vii) Verification of Software Problem Resolution*. This task verifies if the problem has been solved, trends have been inverted, and change requests have been implemented without adding further issues.

Finally, *(viii) Test Documentation Content* is generated. These reports usually include test results, found faults, tested software version, etc.

3.5.5 Software Requirements for Medical Device Development

First of all, device software classification must be identified, as depending on this, the scope of other requirements may vary. Establishing a software development plan covering all design and development phases is also necessary.

The software design process must start with the extraction of software requirements. These must be transformed into an architectural design and then a detailed one.

Once the design is in place, the software must be implemented, and the different components and their integration must be verified. The results of the verification must be recorded.

It will also be necessary to check that the system meets the requirements. Once this verification has been carried out and the required documentation has been generated, the software can be released.

Regul. Des. Req. 9 Identify software safety classification.
Regul. Des. Req. 10 Software Development Planning.
Regul. Des. Req. 10.1 Plan design and development, verification, risk management, documentation and configuration management (Class A, B, C).
Regul. Des. Req. 10.2 Document and keep updated software development plan (Class A, B, C).
Regul. Des. Req. 10.3 Plan software integration and integration testing, items to control, identification and avoidance of common software defects (Class B, C).

Regul. Des. Req. 10.4 Plan software development standards methods and tools (Class C).

Regul. Des. Req. 11 Software Requirement Analysis.

Regul. Des. Req. 11.1 Document and update software requirements (Class A, B, C).

Regul. Des. Req. 11.2 Re-evaluate risk analysis (Class A, B, C).

Regul. Des. Req. 11.3 Verify software requirements (Class A, B, C).

Regul. Des. Req. 11.4 Include risk control measures in software requirements (Class B, C).

Regul. Des. Req. 12 Architectural design.

Regul. Des. Req. 12.1 Transform requirements into architecture (contemplate interfaces, functional and performance requirements, SOUP items) (Class B, C).

Regul. Des. Req. 12.2 Verify software architecture (Class B, C).

Regul. Des. Req. 12.3 Identify segregation necessary for risk control (Class C).

Regul. Des. Req. 13 Detailed design.

Regul. Des. Req. 13.1 Divide software into software units (Class B, C).

Regul. Des. Req. 13.2 Develop a detailed design for each software unit (including interfaces) (Class C).

Regul. Des. Req. 13.3 Verify detailed design (Class C).

Regul. Des. Req. 14 Software unit implementation and verification.

Regul. Des. Req. 14.1 Implement each software unit (Class A, B, C).

Regul. Des. Req. 14.2 Establish unit verification process, and define acceptance criteria (Class B, C).

Regul. Des. Req. 15 Software integration and integration testing.

Regul. Des. Req. 15.1 Integrate and verify software units, and contemplate regression tests (Class B, C).

Regul. Des. Req. 15.2 Record integration tests (Class B, C).

Regul. Des. Req. 16 Software system testing.

Regul. Des. Req. 16.1 Establish tests for software requirements (Class A, B, C).

Regul. Des. Req. 16.2 Retest after changes (Class A, B, C).

Regul. Des. Req. 16.3	Record system testing (Class A, B, C).
Regul. Des. Req. 17	Software release.
Regul. Des. Req. 17.1	Document released versions (Class A, B, C).
Regul. Des. Req. 17.2	Document known residual anomalies (Class A, B, C).
Regul. Des. Req. 17.3	Ensure verification is complete (Class A, B, C).
Regul. Des. Req. 17.4	Evaluate known residual anomalies (Class B, C).
Regul. Des. Req. 18	Perform software risk analysis according to ISO 14971 (see Figure 3.6-1).
Regul. Des. Req. 19	Software configuration management: identification of software units and SOUPs.

3.6 ISO 14971 - Medical Device Risk Management

ISO 14971 [213] provides a framework for managing the risks associated with a medical device. For this purpose, it defines patient, operator, and human risk management processes. More specifically, this standard was developed to enable manufacturers of medical devices to create and maintain a risk management system.

The ISO 14971 standard is based on EN 1441, Medical devices - Risk Analysis, published by the European Committee for Standardization (CEN) in 1997 and the ISO 14971-1, Medical devices - Risk management - Part 1: Application of risk analysis, published in 1998. The first edition of ISO 14971 was published in 2000, the second in 2005 and the third in 2019 [224]. Since 2021, this standard has been harmonised with the two European Regulations, 2017/745 (MDR) and 2017/746 (IVDR) [224].

3.6.1 Structure of ISO 14971

The ISO 14971 standard is divided into five parts. The following is a summary of the content of the standards:

- Part 1 – Scope: This section describes the scope of the regulation.

- Part 2 – Normative references: ISO 14971 does not contain references to any standards.
- Part 3 – Terms and definitions: It defines the main terminology used in the standard. The main concept, **risk**, is defined as *the combination of the probability of damage occurrence and its severity*. **Risk management** is the *systemic application of policies, procedures and management practices for the analysis, assessment, control and monitoring of risk*.
- Part 4 – General requirements and risk management processes: It defines the general processes of risk management, analysis, assessment and control. It also covers the evaluation of the acceptability of the overall residual risk, the generated reports and the production and post-production information.
- Part 5 – Annexes: Additional information such as justification of requirements, an overview of the risk management process, examples of hazards, foreseeable sequences of events and hazardous situations are included in annexes.

3.6.2 Scope of ISO 14971

This standard defines processes for a medical device manufacturer to identify, estimate, evaluate and control the hazards associated with their product. This standard applies to medical devices, including in vitro diagnostic kits. In addition, ISO 14971 applies to all stages of the life cycle of a medical device. This standard does not apply to clinical decision-making or define acceptable risk levels.

Although the standard does not require the manufacturer to have a quality management system such as ISO 13485, risk management can be part of a QMS.

3.6.3 General requirements and risk management processes of ISO 14971

The standard defines general requirements that must be met for efficient medical device risk management. Firstly, it determines that a manufacturer must establish, document and maintain a continuous risk management process throughout the device's life cycle. This process should include risk analysis, assessment and

control, and production and post-production information. Figure 3.6-1 shows a graphical diagram of the risk management process.

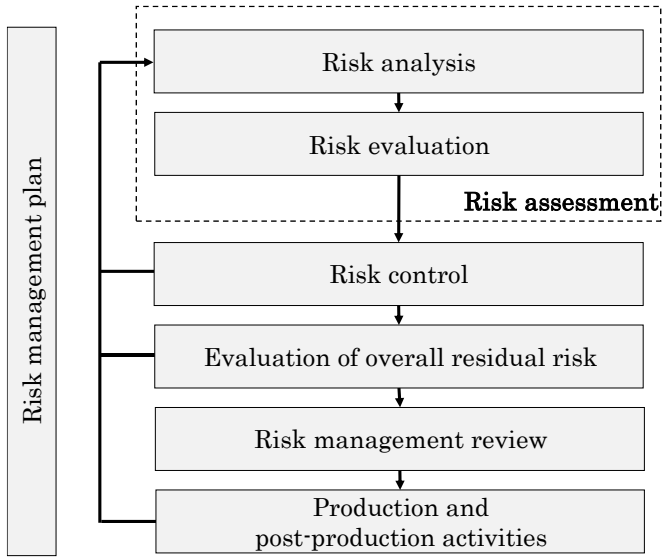


Figure 3.6-1: Main components of the risk management process.

Firstly, the standard defines that the manufacturer has to establish and document a risk management plan. It is also required to have and maintain a risk management file. This file must provide traceability for each identified risk by defining performed analysis, its evaluation, the implementation and verification of control measures and the absence of residual risk.

The standard defines **top management** as *the person or group who directs and controls the manufacturer at the highest level and must be committed to the risk management process by ensuring adequate resources*. Furthermore, top management is responsible for defining the policy for determining risk criteria and periodically reviewing the risk management process to ensure its effectiveness and adequacy.

Once risk management has been planned, all the processes defined in Figure 3.6-1 must be undertaken.

- Risk analysis:

During risk analysis, all activities and results must be recorded in the risk management file. Firstly, the *(i) Intended Use and*

Reasonably Foreseeable Misuse must be documented. The manufacturer must identify and report the qualitative and quantitative characteristics that could affect the product's safety.

Then, the *(ii) Identification of Safety-related Characteristics* must be made. The manufacturer must document the known and foreseeable hazards of the product in the risk management file.

Finally, the *(iii) identification of hazards and hazardous situations* must be carried out. For this, it is necessary to consider the foreseeable sequences or combinations of events that may result in hazardous conditions.

- Risk evaluation:

After risk analysis, a risk evaluation must be carried out for each identified hazardous situation; the manufacturer must decide if a risk mitigation strategy is required. The results must be recorded in the risk management file.

- Risk control:

This process carries out for tasks when risk mitigation action is required.

First, the *(i) Risk Control Analysis* is done. The manufacturer should identify appropriate risk control measures and use one or more options. These include safety by design, health product or manufacturing process protections, and safety information.

This analysis will be followed by the *(ii) Implementation of Risk Control Measures*. Performance should be linked to the verification of their effectiveness.

Once the risk mitigation measures have been established, the *(iii) Residual Risk Evaluation* is necessary. If the residual risk is acceptable, the manufacturer shall decide which residual risks are published in the product documentation. If the risk is considered unacceptable, a control mechanism must be implemented.

When controlling an additional risk using any criteria set out in the risk management plan is not feasible, the manufacturer must perform *(iv) a Benefit-Risk Analysis*. Using available public

evidence, studies and reports, it must demonstrate that the benefits outweigh the residual risk. If this is not met, the identified risk is considered unacceptable.

Furthermore, *(v) Risk Arising from Risk Control Measures* must be reviewed to avoid introducing new risks.

Finally, the manufacturer must *(vi) Ensure the Completeness of Risk Control,* guaranteeing that all identified hazards have been considered. All the described processes have to be included in the risk management file.

- Evaluation of overall residual risk:

Once the risk control process has been completed, residual system risks should be assessed. As in the risk control process, whether the residual risks are acceptable should be identified. If so, proceed with the steps described above.

- Risk management review:

The manufacturer must undertake a risk process review and ensure an adequately implemented risk management plan, acceptable overall residual risk, and established mechanisms for obtaining production and post-production information. All these must be done to sell the medical device.

- Production and post-production activities:

Finally, the manufacturer must establish, document and maintain a mechanism for collecting and reviewing medical device information in the production and post-production stages.

3.6.4 Risk Management Requirements for Medical Device Design and Development

A risk analysis of all system components will be necessary. Following this analysis, the risks shall be assessed. Control measures must be implemented if a risk is unacceptable to the manufacturer.

The risk analysis is considered an iterative process as it shall be repeated until the residual risk, the risk present in the device after

applying the control measures, is considered acceptable by the manufacturer.

Regul. Des. Req. 20	Risk analysis.
Regul. Des. Req. 20.1	Document intended use and reasonably foreseeable misuse.
Regul. Des. Req. 20.2	Identification of characteristics related to safety.
Regul. Des. Req. 20.3	Identification of hazards and hazardous situations.
Regul. Des. Req. 21	Risk evaluation: define risk mitigation strategy for each identified hazardous situation.
Regul. Des. Req. 22	Risk control.
Regul. Des. Req. 22.1	Identify risk control measures.
Regul. Des. Req. 22.2	Implementation of risk control measures.
Regul. Des. Req. 22.3	Residual risk evaluation.
Regul. Des. Req. 22.4	Benefit-risk analysis.
Regul. Des. Req. 22.5	Review risk arising from risk control measures.

3.7 IEC 62366 - Medical Device Usability

The IEC 62366 standard refers to usability engineering for medical devices. This standard is a process-based standard to allow medical device manufacturers to design products with high usability.

The number of medical devices for patient observation and treatment with user interfaces has increased in recent years. Integrating these new functionalities has increased the number of errors in their use. The lack of design simplicity or the difficulty in learning to use them is often the source of these errors. Usability engineering, the design of a user interface according to processes that guarantee adequate medical product safety, is the key. Therefore, this standard aims to minimise the risks related to user interface design in medical equipment.

This standard has two parts, IEC 62366-1 [225] and IEC 62366-2 [226]. Its first version was published in 2007 by the IEC, the International Electrotechnical Commission. In 2015 it was updated,

resulting in IEC 62366-1, Application of usability engineering to medical devices. In 2016, IEC 62366-2, Guidance on applying usability to engineering to medical devices, was generated. The requirements and tasks to be fulfilled to achieve a usable medical device are defined in IEC 62366-1 [214]. The IEC 62366-2 [214] is only a guide for correctly applying IEC 62366-1 [227].

The IEC 62366 standard is accepted in both the European Union and the United States, so manufacturers from these markets can rely on this specification to develop their devices.

3.7.1 Structure of the IEC 62366-1 standard

The IEC 62366-1 standard consists of 5 parts:

- Part 1 – Scope: It defines the scope of the standard.
- Part 2 – Normative references: This standard refers to ISO 14971, Medical devices – Application of risk management to medical devices.
- Part 3 – Terms and definitions: This chapter defines the most relevant terms of the standard. As a primary term, **usability** is defined as *the characteristic of the user interface that facilitates use and thereby establishes effectiveness, efficiency, and user satisfaction in the intended use environment*. Another term, **usability engineering**, is defined as *the application of knowledge about human behaviour, abilities, limitations, and other characteristics to the design of medical devices (including software), systems and task to achieve adequate usability*.
- Part 4 – General requirements and usability engineering process: It defines the general conditions of usability engineering and the associated processes.
- Part 5 – Annexes: Additional information is included in annexes, such as the general guidance and rationale, examples of possible hazardous situations related to usability or evaluation of a user interface of unknown provenance.

3.7.2 Scope of IEC 62366-1

IEC 62366-1 specifies the process for analysing, specifying, designing, verifying and validating the usability of a medical device under normal use conditions. This standard is intended to mitigate the risks caused by usability problems of a device. IEC 62366-1 can be used to identify but not to assess or mitigate risks associated with the abnormal use of the device, that is, when it is not used following the device's instructions. In this case, the assessment is performed according to ISO 14971. This standard does not apply to clinical decision-making regarding medical devices.

This standard is oriented to minimise the risks related to interface design. Therefore, it refers to ISO 14971, specifying that in the case of following the processes defined in this standard, unless there is objective evidence, the residual risks that appear after the execution of the described procedures is acceptable.

3.7.3 Categories of user actions

The user's actions can be categorised into two main groups, normal use and abnormal use:

- Normal use:

It includes routine inspection and adjustments by any user according to the instructions for use or under generally accepted practice for those medical devices provided without instructions. At the same time, normal use can be categorised into Correct use and Use Error. Correct Use is when the device is normally carried out without errors.

Use Error is the error identified during Normal Use of the device. These errors can be caused by perception error (failure to see visual information, hear auditory details, etc.), cognition error (memory failures, knowledge failures, etc.) or action error (inappropriate force applied to components, contact with faulty parts, etc.).

- Abnormal use:

Abnormal use is a conscious, intentional act or intentional omission of an action that contradicts or violates regular use. It occurs when

an exceptional violation, sabotage, deliberate disregard for the contraindications or reckless use is done.

3.7.4 General requirements and usability engineering process of IEC 62366-1

The manufacturer must establish, document, and maintain a usability engineering process that addresses user interactions with the medical device during the transport, storage, installation, operation, maintenance, repair and disposal. The results of this process shall be recorded in the usability engineering file. This file shall collect all records associated with usability engineering processes and form part of the overall project plan.

The usability engineering process is divided into ten tasks which are described below:

- Prepare use specifications:

First, it is necessary to define the use of the device; for this, it is recommended to specify, at least, the intended medical indication, patient population, part of the body interacted with, and user profile and environment.

- Identify user interface characteristics related to safety and potential use errors:

Next, safety-related specifications according to ISO 14971 must be identified. Based on the characteristics of the user interface and the defined specification, usage errors that may occur have to be placed.

- Identify known foreseeable hazards and hazardous situations:

The manufacturer must also identify known or foreseeable hazards relating to the device's usability. As detailed in ISO 14971, the sequence of possible hazards and the severity of the damage must be identified. For this identification, the specification of the application, the user profile, the context of use and the current information on potential usage errors must be considered.

- Identify and describe hazard-related use scenarios:

The manufacturer must identify the use cases in which a hazard may exist. Tasks, error sequences and severity of possible damage must be determined for each use case.

- Select the hazard-related use scenarios for summative evaluation:

The summative evaluation is the user interface evaluation conducted at the end of the development. It aims to obtain objective evidence that the user interface can be used safely.

In this task, the manufacturer shall select the use scenarios to be included in the summative evaluation. For that, the manufacturer may check all use cases or a group of them; this can be determined based on the severity of the possible damage.

- Establish user interface specification:

User interface specification should consider the definition of the use cases, potential use errors and identified hazardous situations. The specification should include testable requirements, documentation, and training for using the device.

- Establish user interface evaluation plan:

Once the specifications have been defined, the user interface evaluation plan must be specified, requiring documenting and identifying the formative and summative evaluation methods.

The formative evaluation explores the user interface's strengths, weaknesses and unanticipated use errors. In contrast, summative evaluation is intended to determine whether the interface is safe to use.

The formative evaluation plan must include the acceptance criteria, usability objectives, test environment, methods and the definition of the techniques used. The summative evaluation plan should include acceptance criteria, test environment, methods and techniques, participants and satisfaction rates.

If usability tests are to be carried out, they must be documented. The evaluation can be both qualitative and quantitative and must carry out in different environments of use, such as in laboratories or simulated scenarios.

- Perform user interface design, implementation and formative evaluation:

After defining verification plans, the implementation of the interface and the documentation generation is performed.

Finally, the formative evaluation is carried out, and the evaluation data must be recorded in the usability engineering file. If a new risk is identified, the tasks described in this section must be implemented again.

- Perform a summative evaluation of the usability of the user interface:

After the design, implementation and formative evaluation, the manufacturer must complete the summative evaluation. For this, the manufacturer can use information obtained from summative evaluations of products with equivalent user interfaces; using this data requires the performance of a technical rationale.

The summative evaluation results are analysed to identify if any damage can occur due to device misuse. If so, the damage source has to be determined; if necessary, the steps defined for mitigating usability risks must be applied again. In contrast, the device has an acceptable usability safety level if all these processes can be completed successfully.

- User interface of unknown provenance (UOUP):

The user interface to be evaluated may be of unknown origin. For example, an already developed interface must be integrated into the medical product. In such cases, the process to be followed is slightly different.

The first step is to define the interface use specification and then review UOUP post-production information to identify existing issues or any other relevant information.

Hazards and hazardous situations related to usability must also be identified and risk control performed according to ISO 14971. Finally, the residual risk should be assessed, and a decision on the usability of the UOUP should be made.

Figure 3.7-1 summarises an example of a usability engineering project, detailing each process's phases.

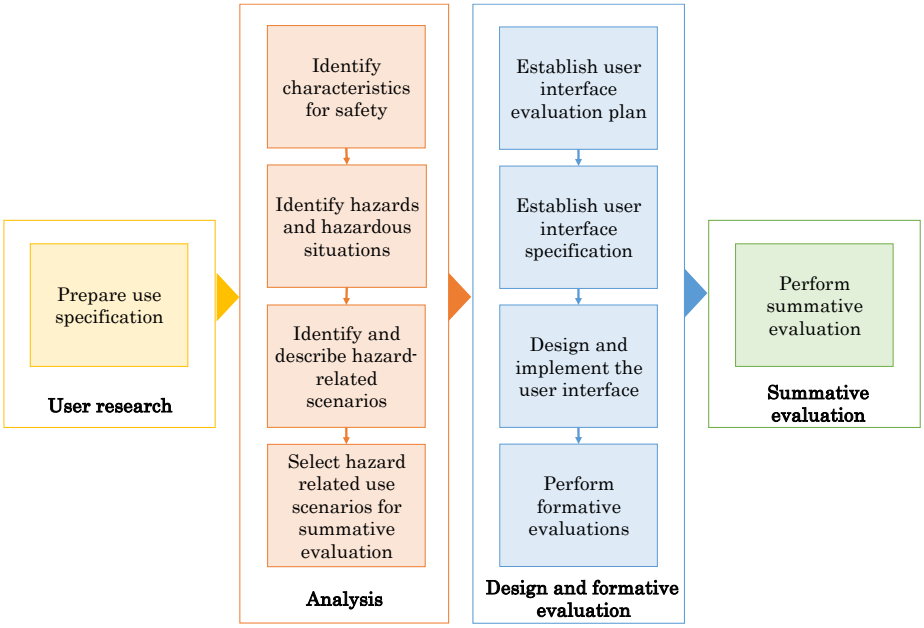


Figure 3.7-1: Example of a usability engineering project.

3.7.5 Usability Requirements for Medical Device Development

The usability requirements are based on a risk analysis of use. For this, defining the device's use case and performing a risk analysis will be necessary.

In addition, this standard requires summative and formative evaluation to assist in the design and development phases. The formative evaluation explores the user interface's strengths, weaknesses, and unanticipated use errors. In contrast, summative evaluation is intended to determine whether the interface is safe to use.

Regul. Des. Req. 23	Define use specification (identify characteristics for safety).
Regul. Des. Req. 24	Perform a usability risk analysis using ISO 14971 (see Figure 3.6-1 and Figure 3.7-1).
Regul. Des. Req. 25	Perform summative and formative evaluations.

3.8 ISO 13485 - Medical Device Quality Management Systems

ISO 13485 [215] is a standard that specifies the requirements of a quality management system for medical device manufacturers.

This standard was published in 1996 by the International Organization for Standardization (ISO); this first version was based on ISO 9001:1994. In 2003, after the update of ISO 9001, ISO 13485 was updated. The next update was made in 2012. The version published in 2016 is in force and is based on ISO 9001:2008 [228].

This process-based standard covers all stages of the product life cycle: design, development, production, storage, distribution, installation, technical support, de-installation and final disposal. It also covers the design, development and service supply of medical devices. ISO 13485 can be adopted by both medical device manufacturers and also by their suppliers.

This standard is the main quality system for medical devices in Europe, Canada and Australia. It also serves as the basis for compliance with quality systems in countries such as Japan, Korea and Brazil [228]. It has recently been published that the FDA intends to use ISO 13485 as the basis for its quality systems legislation [229].

ISO 13485 is based on ISO 9001; however, this standard includes some particular requirements for organisations involved in the life cycle of medical devices and excludes some requirements of ISO 9001 that are inappropriate as regulatory requirements.

This standard does not include specific requirements for other management systems such as environmental management, health and safety at work, financial management, etc. However, aligning

this quality management system with other existing ones is possible.

3.8.1 Structure of the ISO 13485

ISO 13485 consists of the following five parts:

- Part 1 – Scope: It defines the scope of the standard.
- Part 2 – Normative references: This standard refers to ISO 9000 - Quality management systems, fundamentals and vocabulary.
- Part 3 – Terms and definitions: This section defines the most relevant terms of the standard. **Process**, as the primary term introduced from ISO 9000, is defined as *the set of activities that are linked to each other and use inputs to provide an intended result*. **Procedure** is also defined as *the specific way of carrying out an activity or process*.
- Part 4 – General requirements and quality management system processes: The general requirements of the quality management system are defined, and the related tasks or items are defined.
- Part 5 – Annexes: Additional information to correlate ISO13485:2003 and ISO13485:2016, and ISO 13485:2016 and ISO 9001:2015 is included in the annexes.

3.8.2 Scope of ISO 13485

This standard specifies the requirements that a quality management system must meet when an organisation needs to demonstrate its capability to provide medical devices and related services in compliance with applicable customer and regulatory requirements.

ISO 13485 applies to organisations involved in one or more stages of the life cycle of a medical device. It may also be used by suppliers providing product parts to such organisations.

When some applicable processes are carried out outside the medical device manufacturer, the manufacturer must incorporate them into its quality management system through process monitoring, maintenance and control.

3.8.3 Requirements and quality management system processes

The standard defines the general requirements and the associated documentation. Furthermore, it describes processes for implementing the quality system: management responsibility, resource management, product realization and measurement analysis and improvement.

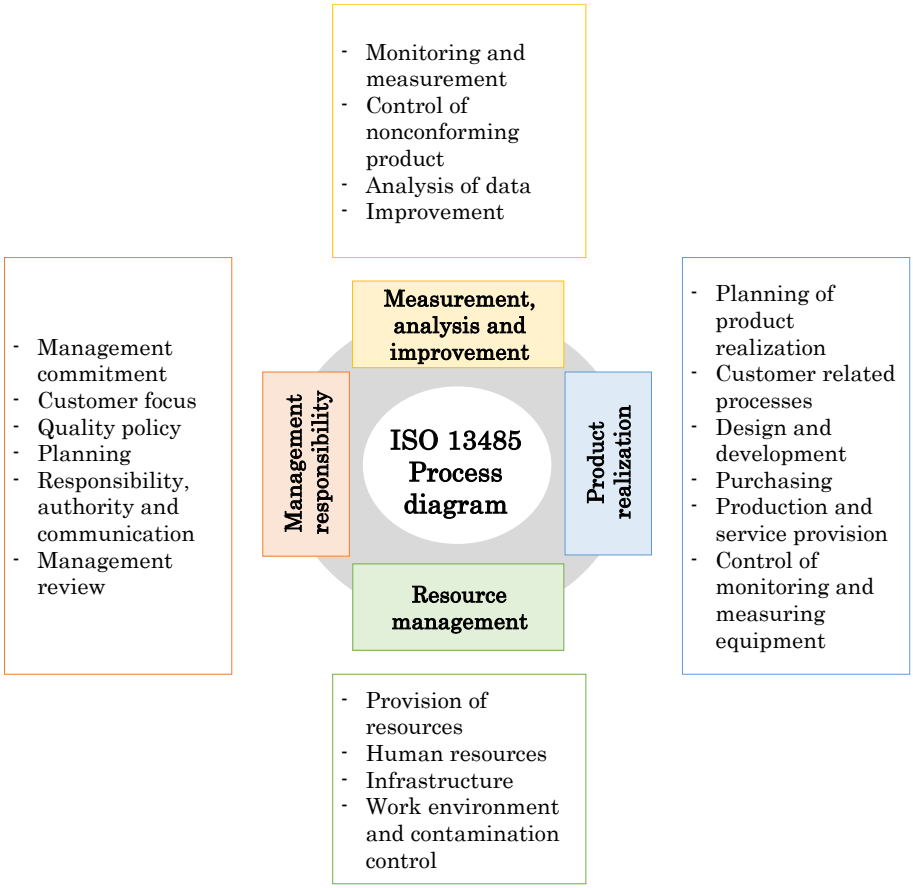


Figure 3.8-1: ISO 13485 quality management system's process diagram.

3.8.3.1 General requirements

The organisation has to document a quality management system and maintain its effectiveness according to the requirements of ISO 13485 and the applicable regulatory requirements. It is also required to register the role played by the organisation.

ISO 13485 outlines identifying processes for the quality management system to determine the sequence and interaction between them and establish criteria and methods for ensuring effective process operation and control. It also requires ensuring the availability of resources and information, monitoring, measuring and analysing processes and implementing the necessary actions to achieve the planned results.

In outsourced processes, these need to be identified within the management system and appropriate controls established.

- Documentation requirements:

In terms of documentation, it will be required to generate and include the following in the quality system (i) quality policies and objectives, (ii) quality manual, (iii) documented procedures and records, (iv) documents to ensure the effective planning, operation, and control processes and (v) other documentation specified by applicable regulatory requirements.

The *quality manual* shall include the scope of the quality management system, documented procedures, and a description of the interaction between processes. It will also set out the structure of the QMS documentation.

Furthermore, the organisation must establish a document called the *medical device file* for each type of medical device. This document includes the description of the device, its purpose, labelling, specifications, etc.

Another critical aspect of the standard is the *control of documents*. It will be necessary to review and approve product documentation, review and update them, and, when necessary, they have to identify the current revision status. It will also require preserving manufacturing and testing documents during the product's life.

Finally, it is necessary to perform the *control of records*. Records are a particular type of document that must be kept for at least the life of the device or at least two years after the product's release. A documented procedure for both document control and records control will be necessary.

- Management responsibility:

According to the standard, it is essential to have *(i) Top Management Commitment* to the QMS processes, providing evidence of its engagement in developing, implementing and maintaining the effectiveness of the QMS.

Top management shall provide *(ii) Customer Focus*, ensuring that regulatory and customer requirements are met.

Management shall also ensure an effective *(iii) Quality Policy* aligned with the organisation's needs and assures compliance, continued suitability and understanding of all the organisation's staff.

Furthermore, top management will ensure an effective *(iv) Planning of the QMS* by setting clear objectives to meet customer and regulatory requirements.

Top management must undertake the *(v) Responsibility, Authority, and Communication of the Standard* by defining and documenting authorities and responsibilities and designating a member of management responsible for the system. It must also ensure appropriate communication within the organisation to ensure the effectiveness of the QMS.

The *(vi) Management Review of the QMS* is a critical point in the standard; this ensures that the system's suitability, adequacy, and effectiveness are maintained over time. These reviews should be recorded, and a documented procedure for management review is required.

3.8.3.2 Resource management

The standard defines that the *(i) Provision of Resources* is necessary to comply with legal and regulatory requirements, implement the QMS and maintain its effectiveness.

On the one hand, it requires having competent, trained, skilled and experienced *(ii) Human Resources*. In addition, it specifies documenting competency processes such as skills development.

On the other hand, the organisation must document the *(iii) Infrastructure Requirements* to achieve conformity with the product requirements. For example, buildings, work areas, hardware and software processes, equipment and support services, among others, must be defined.

Finally, it is necessary to have a documented procedure for *(iv) Work Environment and Contamination Control*. Requirements for health, cleanliness, work environment, clothing and contamination control must be defined.

3.8.3.3 Product realization

According to ISO 13485, product realisation requires the execution of 6 phases: planning of product realization, customer-related processes, design and development, purchasing, production and service provision, and control of monitoring and measuring equipment.

- Planning of product realization:

First, the planning of product realization must be carried out. The organisation should plan the product realization by determining the quality and product requirements, establishing specific processes and resources, verification activities, validation, monitoring, measurement, inspection, testing, distribution, storage and traceability. It will also be necessary to document processes for product development risk management.

- Customer-related processes:

According to ISO 13485, customer-related processes must be documented. Customer specifications, including delivery and post-market, must be identified and reviewed for this. In cases where the customer does not provide a clear definition, it is necessary to confirm these requirements before acceptance. Finally, the organisation should plan and document communication with the end user. This plan has to contemplate product information, enquiries, contracts, notices and customer feedback.

- Design and development:

For the design and development phase, documented procedures must be established. This phase is divided into Planning, Inputs, Outputs, Review, Verification, Validation, Transfer, Change control and Files.

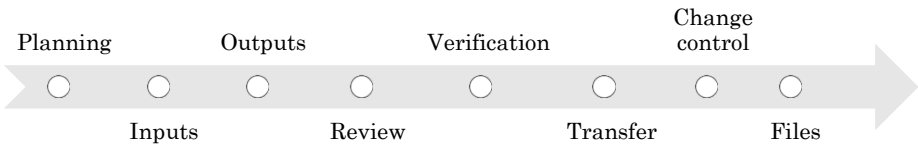


Figure 3.8-2: Design and development process phases

The *(i) Design and Development Planning* should cover design and development stages, critical reviews, verification, validation and transfer activities. It also includes responsibilities and methods to ensure the required resources and their traceability.

The *(ii) Inputs* and *(iii) Design and Development Outputs* should be documented. The inputs should define the functional, safety and regulatory requirements. The outputs should provide appropriate information for purchasing activities, production and service provision, reference the acceptance criteria and specify the essential product characteristics for safe and correct use.

During design and development, *(iv) Systematic Reviews* should be conducted to assess compliance with requirements, identify deviations and propose corrective actions.

After the development, the *(v) Verification* has to be performed. This process checks whether the requirements of the input elements have been fulfilled.

Then, the *(vi) Validation of the Design* according to the plan is executed. This validation should ensure that the resulting product meets the requirements for its intended purpose. The validation should be performed on a representative product, such as initial production units or equivalents. As part of the validation process, clinical or performance evaluations of the device should be carried

out according to applicable regulations. This validation should be performed before the product release.

Both verification and validation must be documented and recorded, identifying the methodology used, acceptance criteria, sample size and results.

The manufacturer must document the procedures for design and development *(vii) Transfer to Manufacturing*. It must also be verified that the results are suitable for manufacturing; this process must be done before final product manufacturing.

It is also required to manage the design and development *(viii) Change Control*; this procedure should be documented by identifying the relevance of the change, performance, usability, safety and regulatory requirements.

Finally, the design and development *(ix) File* shall be updated for each product type. This file includes records demonstrating compliance with the design and development requirements and their change history.

- Purchasing:

ISO 13485 also defines the purchasing phase. It is necessary to control suppliers by setting evaluation and control criteria to ensure a good purchasing process. This process should be established according to supplier capability, performance, the effect of the purchasing item on the quality of the medical device and in proportion to the associated risk of the final product.

During the purchasing phase, specifications, acceptance criteria, qualification of the supplier staff and quality management system requirements must be clearly defined. Finally, it has to verify whether the purchased item meets the established acceptance criteria.

- Production and service provision:

The production and service provision phase is one of the most critical phases of the QMS. In this process, it is necessary to control production and service provision using documented procedures and methods. The parameters to be measured must be identified, and a

record of them must be maintained to ensure correct supervision. If monitoring equipment must be used, it has to be correctly calibrated according to plan. Also, when necessary, the qualification of the manufacturing infrastructure must be carried out.

This phase should address and identify product cleanliness and contamination control requirements, medical device installation and verification, service support requirements, and particular product requirements such as sterile medical devices.

Production and service delivery processes must be validated to demonstrate that the organisation can regularly achieve the planned results according to the requirements of the sanitary product to be manufactured.

The product must be identified during all stages of its manufacture. The identification must allow for knowing its status and managing the traceability of the manufacturing process.

Furthermore, the customer's product must be identified, verified, protected and safeguarded while the manufacturer uses it. In case of deterioration or inappropriate use, the customer must be informed. Finally, the organisation must document procedures to preserve product conformity during processing, storage, handling and distribution.

- Control of monitoring and measuring equipment:

Finally, ISO 13485 covers the control of monitoring and measuring equipment. The organisation must define the monitoring and measurement equipment. This equipment must be calibrated or verified according to established calibration procedures. Computer applications used during production and service delivery must also be validated before use.

3.8.3.4 Measurement, analysis and improvement

The organisation shall plan and implement monitoring, measurement, analysis and improvement procedures to demonstrate product and quality management system conformity and maintain its effectiveness.

- Monitoring and measurement:

For monitoring and measurement purposes, the organisation should collect and follow up on feedback received from the customer. It shall set up a procedure for dealing with customer complaints. It shall also establish a procedure to communicate to regulatory bodies the identified issues.

Additionally, it is required to conduct periodic evaluations through internal audits to ensure that the defined QMS is working effectively. Processes and products must also be monitored and measured to verify that they are fulfilling their purpose.

- Control of nonconforming product:

It is also necessary to manage non-conformities by carrying out actions to identify them both before and after product release.

In case rework is required, that is, the execution of any action that involves performing a company process again, the products must be subjected to the same verifications and approvals as the original process.

- Analysis of data:

In terms of data analysis, appropriate data must be determined, collected and analysed to provide information on the system's suitability, adequacy and efficiency. Typical data are feedback, conformance to requirements, process and product characteristics and trends, suppliers, audits and technical assistance reports.

- Improvement:

Finally, improvement processes should identify and implement changes to ensure and maintain the continuous suitability, adequacy and effectiveness of the QMS. To this end, both corrective and preventive actions should be taken. Corrective actions should be used to eliminate the causes of non-conformities. In contrast, preventive measures should eliminate the causes of potential non-conformities.

3.8.4 Quality Management Requirements for Medical Device Design and Development

This standard requires a plan for product development. Specifically, the design and development phases describe the need to plan the process, define the design's inputs and outputs, and conduct systematic design reviews, verifications, and validations. It also contemplates the transfer phase to manufacturing.

Additionally, it details the relevance of controlling component purchases and measuring equipment during the prototyping and production phases. According to ISO 13485, it will also be necessary to document procedures, reviews, and controls.

Regul. Des. Req. 26	Documentation.
Regul. Des. Req. 26.1	Document procedures and records.
Regul. Des. Req. 26.2	Review and approval of documentation.
Regul. Des. Req. 26.3	Control traceability of documentation.
Regul. Des. Req. 27	Planning of product realization.
Regul. Des. Req. 28	Design and development.
Regul. Des. Req. 28.1	Planning: it should cover reviews, verification, validation, transfer, responsibilities, and methods to ensure the traceability of resources.
Regul. Des. Req. 28.2	Input and outputs: define requirements (functional, safety and regulatory) and outputs (information for purchasing, production, acceptance criteria, etc.).
Regul. Des. Req. 28.3	Systematic reviews: monitor progress and establish corrective actions if necessary.
Regul. Des. Req. 28.4	Verification: perform and document verification and check if requirements have been fulfilled.
Regul. Des. Req. 28.5	Validation: perform and document validation on a representative product (it contemplates clinical or performance evaluations, EMC...).
Regul. Des. Req. 28.6	Transfer to manufacturing: verified development is suitable for manufacturing.

Regul. Des. Req. 29	Control the purchases made during the prototyping phase.
Regul. Des. Req. 30	Control monitoring and measuring equipment.

3.9 IEC 81001-5-1 - Medical Device Cybersecurity

Medical devices are increasingly connected to allow agile data exchange between the device and the hospital network. Currently, many medical devices on the market are not designed according to cybersecurity requirements, as regulations were not directly addressing this potential risk.

In Europe, the Medical Device Directive (93/42/EEC) [230] published in 1993 merely featured a sentence that indirectly referred to cybersecurity aspects. It was not until the new Medical Device Regulation (2017/745), in force since 26 May 2021, that direct reference was made to cybersecurity.

Specifically, the list of harmonised standards for the MDR refers to 3 different cybersecurity standards, for which the date of adoption is May 2024.

- IEC 80001-1[231]: Safety, effectiveness and security in the implementation and use of connected medical devices or health software - Part 1: Application of risk management, which is oriented towards security criteria for networks in which medical devices are embedded.
- IEC 81001-5-1[216]: Health Software and health IT systems safety, effectiveness and security - Part 5-1: Security - Activities in the product lifecycle, focused on the cybersecure design of medical devices. It defines the cybersecure development life cycle.
- IEC/TR 60601-4-5[211]: Medical electrical equipment - Part 4-5: Guidance and interpretation - Safety-related technical security specifications. It is a technical report that complements IEC 81001-5-1 with additional security requirements.

Since 2014, the FDA has published several guidelines to ensure cybersecurity in its equipment. In 2018, the FDA recognised UL 2900 as the first cybersecurity standard [232]; however, this standard has lost momentum in future guidance revisions.

The latest guidance published by the FDA in April 2022, "Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket Submissions", reduces UL 2900 to a cybersecurity testing guideline. On the other hand, it refers to ISA 62443-4-1 as a possible framework for developing the software lifecycle in a cyber-secure way [233].

IEC 81001-5-1 is based on ISA 62443-4-1, providing a standard reference for both markets [234]. However, there are still some differences in the scope of some concepts presented in the FDA guidance and IEC 81001-5-1. Therefore, it is expected that IEC 81001-5-1 will become the reference standard for medical device cybersecurity for Europe and international markets in the coming years.

3.9.1 Structure of the IEC 81001-5-1

The IEC 81001-5-1 standard is divided into five parts. This standard structure is equivalent to that of IEC 62304.

- Part 1 – Scope: The scope of the standard is defined.
- Part 2 – Normative references: This standard does not contain any normative references.
- Part 3 – Terms and definitions: It defines the main terms referred to in the standard. One is **health software** which is defined as *software intended to be used specifically for managing, maintaining, or improving health of individual persons, or the delivery of care, or which has been developed for the purpose of being incorporated into a medical device*. It also defines **Threat** as *potential violation of security, which exists when there is a circumstance, capability, action, or event that could branch security and cause damage to confidentiality, integrity, availability or information assets*. **Trust boundaries** as an *element of threat model that depicts a boundary where authentication is required or a change in trust level occurs*.

- Part 4 – Requirements, processes and tasks: It describes the requirements related to the QMS, risk and safety classification of software. It also covers medical device software's design, maintenance, risk management, and troubleshooting processes.
- Part 5 – Annexes: This includes further information detailed in the standard.

3.9.2 Scope of IEC 81001-5-1

This standard defines the requirements of the software development and maintenance lifecycle to be compliant with the IEC 62443-4-1 (Security for industrial automation and control systems – Secure product development lifecycle requirements).

Based on IEC 62443-4-1, this standard aims to increase the cybersecurity of healthcare software. To this end, it sets out specific activities and tasks.

IEC 81001-5-1 applies to health software, that is, software which is part of a medical device, software as part of hardware specifically intended for health use, software as a medical device and software-only product for other health use.

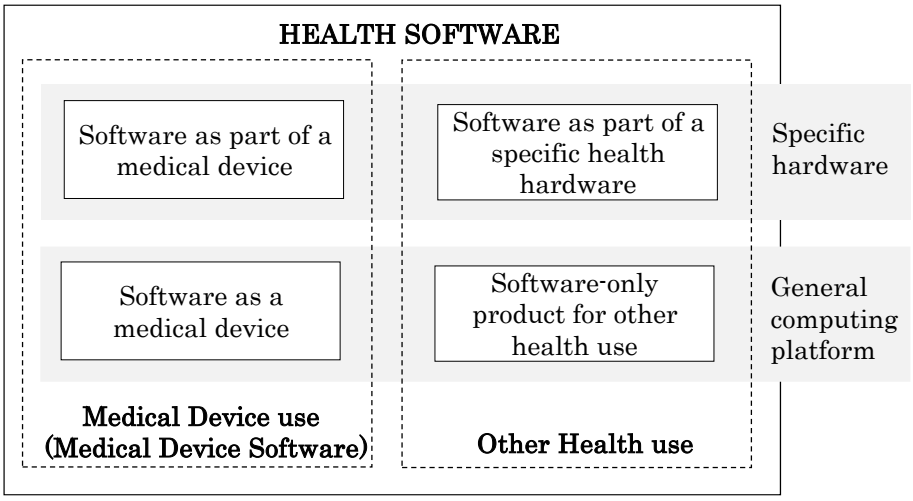


Figure 3.9-1: Health Software definition.

3.9.3 Requirements, processes and tasks

3.9.3.1 General requirements:

This standard defines general requirements for quality management, security risk management and software item classification relating to risk transfer.

Firstly, the requirements of *(i) Quality Management* are established; this defines the need to include in the QMS actions related to security, the identification of responsibility for ensuring security and the applicability of the standard in the different products or parts, including third-party items.

It also defines how to provide security expertise through training or programmes, the continuous improvement of the established processes, the disclosure of security-related issues, the periodic review of security defect management and the documentation of all processes and tasks.

Processes for *(ii) Security Risk Management* must be established. It identifies the need to model threats as a process for identifying vulnerabilities, estimating and assessing threats, controlling them and monitoring the effectiveness of the applied risk control.

Finally, the *(iii) Software Item Classification* relating to risk transfer is defined. Three types of software are identified. The first one is **maintained software,** defined as *software item for which the manufacturer will assume the risk related to security*. The second is **supported software,** defined as *software item for which the manufacturer will notify the customer regarding known risk related to security*. The last one is called **required software**, *software item for which the manufacturer will consider security related risks known before release of the health software*.

The main processes involved in this standard are the same as in IEC 62304, Software Development, Software Maintenance, Software Risk Management, Software Configuration Management and Software Problem Resolution. In the following subsections, these processes will be detailed.

3.9.3.2 Software development process:

The software development process has 8 phases: Planning, Requirements Analysis, Architecture Design, Software Design, Software Unit Implementation and Verification, Software Integration Testing, Software System Testing and Software Release. These 8 phases are the same as those defined in IEC 62304 since this standard fulfils the processes described in IEC 62304 to ensure that the software developed is cybersecure.

The first step is *(i) Software Development Planning*; this involves identifying software security-related activities that apply to each phase of the life cycle, ensuring development environment security and monitoring the IT infrastructure used for development, production or maintenance. It is also essential to establish secure coding standards to guarantee the security of software systems.

Once software development is planned, the *(ii) Software Requirements Analysis* must be performed. These requirements must be documented and cover the installation, operation, maintenance, and decommissioning phases. The requirements have to be reviewed by experts from different fields to ensure their security (architects, developers, verifiers, security advisors, etc.).

Appropriate *(iii) Software Architectural Design* is essential to ensure the security of the software. Defence-in-depth should be considered in each phase of development. Secure design best practices should also be identified, documented, enforced and maintained.

Some best practices are considered: document all trust boundaries, minimise access and user privileges, use proven security software items and simple designs, use secure design patterns, reduce the attack surface, remove backdoors, debug access and document their presence and need to protect them.

Software architectural design must be reviewed and documented. This review must ensure adequate software segregation, follow best practices and control potential security flaws introduced by the architecture.

The performance of *(iv) Software Design* is necessary to develop, document and maintain the secure health software best practices, the application-level software technology and the used programming language.

In addition, measures to address the threats identified in the threat model must be included. The physical and logical interfaces should also be identified and characterised, detailing whether they are externally accessible, their security implications, users, access roles, security capabilities, third-party software and how to mitigate the identified threats. The design should be reviewed and documented.

Once the software is designed, the *(v) Software Unit Implementation and Verification* are performed. The implementation must follow secure coding standards, and a review process must be established.

The review should identify security requirements, use secure coding standards, perform static code analysis, code inspection and walkthroughs, an assessment of the implementation and its traceability and the examination of threats, trust boundaries and assets.

Then, the *(vi) Software Integration Testing* is carried out, considering security policy differences across trust boundaries.

The *(vii) Software System Testing* involves different phases. First, the security requirements testing must be done to verify that the software meets the security requirements and handles error scenarios and invalid inputs. Additionally, threat mitigation testing must be performed to verify the effectiveness of the mitigation for identified threats.

After that, vulnerability testing for identifying and characterizing potential security vulnerabilities. Finally, penetration testing identifies and characterises weaknesses via tests focusing on discovering and exploiting security vulnerabilities. It is also essential to manage the conflicts of interest between testers and developers; in this way, tests can be performed efficiently.

Finally, the *(viii) Software Release* is done. For that, some tasks must be done; the first is the resolution of findings before release.

The release must be documented, including secure operation guidelines, appropriate information about residual risks, process rigour and conformance documentation, including scoping, tailoring and information on coverage documentation.

Also, an integrity verification has to be done for scripts, executables and other files. In addition, a control process must be established to protect private keys used for code signing. Finally, verifying that all security-related issues have been analysed, solved, or addressed is required.

3.9.3.3 Software maintenance process

Software maintenance involves Planning, Problem or Modification Analysis, and Change Implementation.

First, it is necessary to establish *(i) a Software Maintenance Plan*. The manufacturer should define a policy specifying the timeframes for delivering and qualifying product security updates. This policy should contemplate the potential impact of the vulnerability, public knowledge of the vulnerability, whether published exploits exist for the vulnerability, the volume of deployed products that are affected and the availability of an effective external control when a software update is not being provided.

To ensure safety in maintenance, *(ii) Problem and Modification Analysis* must be performed. For this purpose, public incident reports should be monitored. A task should be set up to verify that the manufacturer's security updates address the intended security vulnerabilities and do not introduce unintended effects.

The *(iii) Implementation Modification* consists of three tasks. First, the change must be documented, setting compatibility with the software in question or mitigation alternatives in case an update is impossible. Then, the manufacturer must ensure that security updates are available for maintained software users. Finally, the new version must be provided securely. It is essential to ensure that each applicable update for maintained software allows the user to verify the integrity of the update patch.

3.9.3.4 Security risk management process

The manufacturer must establish and maintain a process for managing software security risks. The risk management process to ensure security can complement the product risk management performed according to ISO 14871.

First, the *(i) Product Security Context* must be documented. This documentation aims to ensure the same understanding of how the product is intended to be used. This document must include location in the network, physical security or cybersecurity provided by the environment where the product will be deployed, isolation, the potential impact on safety caused by security degradation and implemented security controls.

The manufacturer must also carry out the *(ii) Identification of Vulnerabilities, Threats and associated Adverse Impacts* affecting confidentiality, integrity and availability of assets. This process is designed to ensure that each product has a specific threat model for the scope of the product. The threat model should be reviewed and verified by the development team.

It is also necessary to carry out the *(iii) Evaluation of Security risk*. This process should be carried out considering the adverse impact of the risks.

Once the risk has been assessed, the *(iv) Security risk control* has to be considered. It must be ensured that risk control measures are appropriate for reducing security risks to an acceptable level based on security risk acceptance policies. If they are not acceptable, appropriate mitigation actions should be selected, determine whether those actions generate new risks, implement the mitigations, and verify that the measures taken are effective. Residual risk management should be performed in cooperation with product risk management.

Finally, *(v) Monitoring the Effectiveness of the Risk control* process must be executed. The manufacturer should monitor the effectiveness of risk control by collecting information and reviewing it after the software release. Software vendors should also be informed if a software problem is identified in their source code.

3.9.3.5 Software configuration management process

The manufacturer must establish a process of configuration management that includes change control and change history. It must also be able to reproduce a list of included external components that could become susceptible to vulnerabilities.

3.9.3.6 Software problem resolution process

The software problem resolution process involves four steps.

First, a process must be in place to *(i) Receive Notifications* about vulnerabilities through internal and external entities.

Once the vulnerability notifications have been received, the *(ii) Reviewing of the Vulnerabilities* must be done. For this purpose, the applicability of the product, verifiability and related threats must be determined.

Then, the manufacturer must *(iii) Analyse Vulnerabilities* to contemplate their impact on safety, security context, the environment of use, use of defence-in-depth strategy and identify other products or versions containing the security issue.

Finally, the process of *(iv) Addressing Security-related Issues* must execute. The manufacturer has to address security related aspects and determine whether to disclose them and whether and how identified risks will be handled.

Additionally, it must review design and implementation changes for impact on safety, security and effectiveness, inform other processes and third parties of the issue and include periodic reviews of security-related open issues.

3.9.4 Cybersecurity Requirements for Medical Device Design and Development

This standard completes the requirements already defined in IEC 62304 from a cybersecurity point of view. It covers the same design and development phases: Planning, Software Requirement Analysis, Architectural Design, Detailed Design, Software Unit Implementation and Verification, Software Integration Testing, Software System Testing and Software Release.

In this case, this standard emphasises the importance of specifying requirements, design and tests to ensure that the medical device to be developed is cybersecure. To this end, it recommends using software coding standards or good cyber-safe design practices, such as defence-in-depth and effective software segregation [235].

Regul. Des. Req. 31	Classification of each software item.
Regul. Des. Req. 32	Software Development Planning.
Regul. Des. Req. 32.1	Identify security-related items (IT infrastructure for development, production, etc.).
Regul. Des. Req. 33	Software Requirement Analysis.
Regul. Des. Req. 33.1	Document security requirements.
Regul. Des. Req. 33.2	Review security requirements by experts.
Regul. Des. Req. 34	Architectural design.
Regul. Des. Req. 34.1	Document the secure architectural design, contemplate secure best practices, defence in depth and effective software segregation.
Regul. Des. Req. 34.2	Review the architecture design.
Regul. Des. Req. 35	Detailed design.
Regul. Des. Req. 35.1	Document the secure design.
Regul. Des. Req. 35.2	Identify measures to address the identified threads.
Regul. Des. Req. 35.3	Review the design.
Regul. Des. Req. 36	Software unit implementation and verification.
Regul. Des. Req. 36.1	Implement securely (secure coding standards).
Regul. Des. Req. 36.2	Review implementation (security requirements, coding standard, static code analysis, code inspection, traceability, etc.).
Regul. Des. Req. 37	Software integration testing: document and execute.
Regul. Des. Req. 38	Software system testing.
Regul. Des. Req. 38.1	Security requirement testing.
Regul. Des. Req. 38.2	Management of conflicts of interest between testers and developers.
Regul. Des. Req. 39	Software release.
Regul. Des. Req. 39.1	Resolve finding errors.

Regul. Des. Req. 39.2 Document release, including secure operation guidelines.
Regul. Des. Req. 39.3 Integrity verification of scripts, executables, etc.
Regul. Des. Req. 39.4 Verify that security-related issues have been addressed.

Regul. Des. Req. 40 Perform security risk analysis according to ISO 14971 (see Figure 3.6-1).

Regul. Des. Req. 41 Software configuration management: identification of software units and SOUPs.

3.10 Batteries Regulation

More and more medical devices, especially portable embedded devices, include batteries. Different standards exist for regulating the use of batteries in medical devices [236]. These must be considered when selecting the battery to be used. The following standards are identified for the use of batteries in the European medical device market:

- IEC 62133: Secondary cells and batteries containing alkaline or other non-acid electrolytes - Safety requirements for portable sealed secondary cells, and for batteries made from them, for use in portable applications. This standard applies to devices with rechargeable batteries.
- IEC 60086-4: Primary batteries - Safety of lithium batteries. This standard applies to devices with non-rechargeable batteries.

The following certifications are required for FDA-regulated devices:

- UL 2054: UL Standard for Safety Household and Commercial Batteries. It applies to devices with rechargeable and non-rechargeable batteries.
- UL 1642: UL Standard for Safety Lithium Batteries. It applies to devices with rechargeable and non-rechargeable batteries.

In addition, it is essential to mention that the shipment of lithium batteries is considered dangerous and must be appropriately tested

and packaged. The United Nations (UN) regulates the transport of lithium batteries in the UN Manual of Tests and Criteria, Sub-section 38.3. This standard defines all lithium cells and batteries' environmental, mechanical and electrical requirements.

4. Product Design Methodologies

After analysing the regulatory context faced by each device, the need for a design methodology to ease compliance with these regulations becomes evident. Therefore, this chapter reviews existing product design methodologies. It reviews Heavyweight methodologies such as Waterfall, V-model, Incremental, Iterative, Spiral and Prototype model, and Agile methodologies such as Scrum, Kanban and Crystal frameworks. Additionally, Lean Startup, Design Thinking and Stage-Gate are analysed. Hybrid methodologies that combine different methodologies are also discussed. For each of the reviewed methodologies, their suitability or difficulty to be implemented in developing medical devices is detailed. Finally, after finding that none of the state-of-the-art methodologies is adapted to the nature of the development of embedded medical devices and their regulation, each methodology's most relevant requirements or characteristics are extracted. This analysis is intended to provide a starting point for defining a new design and development methodology for embedded medical devices. It is worth highlighting that when defining these requirements, special consideration is given to the requirements that a methodology must meet to enable a start-up company to develop a medical device successfully.

4.1 Heavyweight Methodologies

Heavyweight methodologies are those based on linear and sequential task execution. The basis of these methodologies is to ensure that the development is carried out according to the plan. They base the success of the project on deadlines, costs and quality. The development is planned and divided into different phases, constituting the life cycle. In general, these methodologies tend to have the following characteristics [237],[238]:

- Predictive approach: Classic management defines in detail the expected output and draws up a development plan, which is used to calculate the cost and deadlines. During execution, monitoring and surveillance activities are carried out to avoid any possible deviations from the plan.
- Detailed documentation: Developments tend to have a large amount of documentation as the emphasis is placed on identifying requirements.
- Process-oriented: Its objective is to define a process that can address the needs of the manager, designer, coder, etc. Each process of the methodology is usually divided into different sub-processes.
- Universality: Projects, despite their diversity, have common patterns of execution. Management practices are based on these characteristics and are valid for any project.

In line with these principles, there are different Heavyweight methodologies; the main ones are listed here.

4.1.1 Waterfall methodology

This methodology dates back to 1970 when Winston W. Royce defined the current waterfall model[239]. Waterfall methodology is based on a **sequential product development process**. The project is divided into sequential phases and can only move on to the next step when the previous one has been completed. Its name is derived from the sequentiality this approach requires, as the phases must be placed one on top of the other in a specific order from top to bottom.

This methodology, as presented in Figure 4.1-1, includes 5 phases [240]:

- Requirements:

The first step is establishing the system's needs, which allows good project planning regarding time and cost. The customer will not be able to modify any of the requirements during other phases of the development. This phase includes the generation of the requirements documentation.

- Design:

The overall system architecture and the logical and physical design are defined.

- Implementation:

Development of the associated components is done in this phase. After its execution, the associated software, electronics and mechanics components must be manufactured and documented.

- Verification:

Every developed part is tested to ensure it complies with the client's requirements. After this phase, the development is delivered to the end users so that they can verify it. This task must generate the plans and reports of the system's verifying process.

- Maintenance:

This phase includes installing, commissioning, and correcting errors that may have arisen after the product's release. Generating the required product documentation, such as the user's manual, is also necessary.

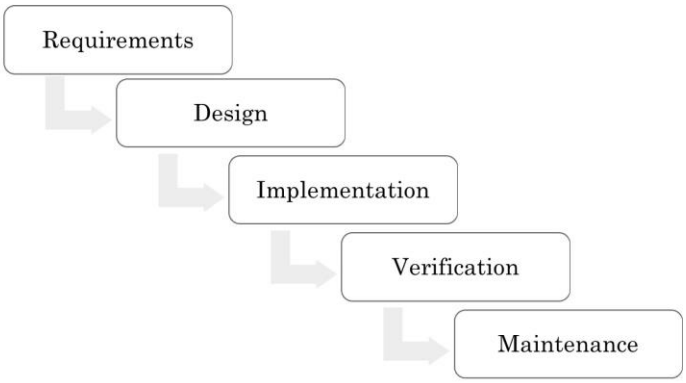


Figure 4.1-1: Waterfall model.

Waterfall's structure is simple and straightforward [241]. Its use is recommended in cases where the requirements are expected to remain stable during development [242]. This method allows planning regarding time and cost during the initial phases of development. In addition, the deliverables generated at each step ensure an efficient flow of information between stages. All this, together with its clear and well-defined structure, makes it a straightforward methodology to control and manage the project. The **clear structure, easy management and the possibility of controlling deviations make it an ideal method for inexperienced teams**.

On the other hand, being a linear and sequential methodology makes it **inflexible to any changes** that may **arise during the development** process [243]. A simple change implies re-executing all the phases, resulting in a costly process.

This methodology **focuses on fulfilling established requirements** and does **not prioritise early customer feedback** [244]. For example, in developments involving software interfaces, this can become a problem since customer feedback is not received until the late stages of the project.

In addition, another difficulty for projects with software components is that **testing is carried out at a very late stage and only at the system level**. Identifying software bugs after the completion of the entire development can lead to a high level of rework [245]. In the case of developing a new medical device, the verification of the critical unit would not be performed until the last phases. It can then conclude that the development is not technically feasible. Keeping

uncertainty until the final stages makes the approach unsuitable for solutions with high technical risks.

This methodology has been widely used for many years, especially in the construction, IT and software development sectors. It is helpful in software development, especially in complex developments, if the solution is divided into small projects. By taking this approach, the team can focus on a specific part and keep complete control of the team's progress.

It is also **appropriate when the development integrates different components**, as in the case of embedded devices. This methodology **allows the team to develop each part independently and integrate** the different elements at the end of each process.

In the case of the development of a new medical device, the verification of critical parts must be completed at the early stages of development. Therefore, this methodology is **unsuitable for developments with high technical uncertainty**. However, the design and development procedures defined in the medical device standards suggest using sequential and linear methods, such as Waterfall, as they detail a specific order to execute some defined processes [246].

> **Methodol. Req. 1** Divide the project into sub-tasks to allow the team to focus on a particular development and to manage the development better.

4.1.2 V-Model methodology

The V-Model methodology is also known as the Verification and Validation Model. It was defined by Paul Rook in 1980 as an evolution of the Waterfall methodology [247]. Like Waterfall, it is a **sequential methodology**, so it requires a complete execution of one phase to move on to the next. In this case, its **focus is more on getting a successful verification of all steps** to ensure a high quality of the development [248].

As presented in Figure 4.1-2, this methodology is based on three processes, the first one, the left part of the V-model that defines the conversion of requirements into design, and the second one, the lower part of the V that contemplates the implementation or

execution. Finally, the third process, the V-model's right part, is the development's integration and verification [249].

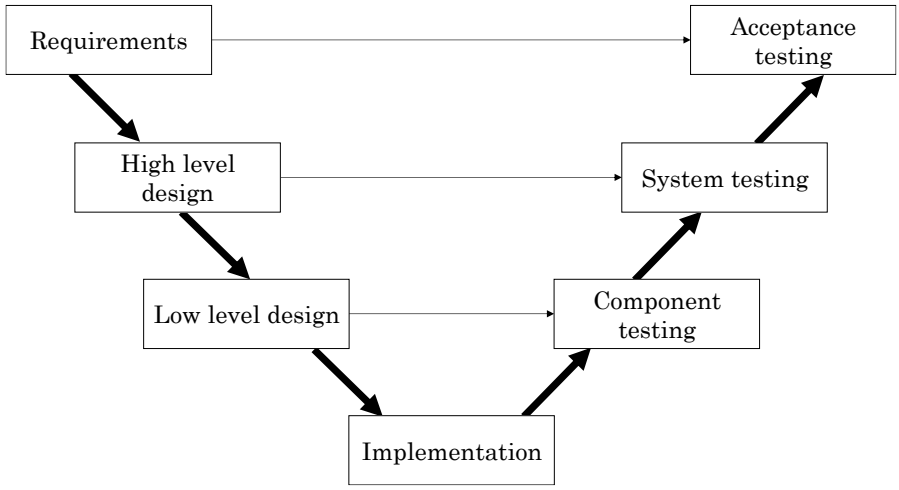


Figure 4.1-2: V-model methodology for product design and development.

Similarly to Waterfall, it is **simple and easy to implement**. In contrast to Waterfall, the planning and definition of the verification are done before development, saving time and cost. On the other hand, the execution of verification is contemplated after implementing the complete system. This methodology **can be inefficient for very complex developments**. Like Waterfall, it is a **rigid methodology without flexibility for changes or errors** [250]. Likewise, this approach does not contemplate using initial prototypes to validate the concept.

This methodology is a**ppropriate for enhancing an existing product with specific functionality** or developing a new generation of an existing device. It is also a methodology to be implemented in teams with high technical knowledge and scenarios with low technological uncertainty.

The V-based methodology is widely accepted in many sectors. Therefore, different evolutions depend on the industry or field of application [251]–[254]. The Verein Deutscher Ingenieure (VDI), the largest German engineering association in Western Europe, introduced it in 2004 under the name VDI 2206 [255] as a development methodology for mechatronic systems. It was accepted

in both scientific and industrial environments [256]. In Figure 4.1-3 VDI 2206 V-model is presented.

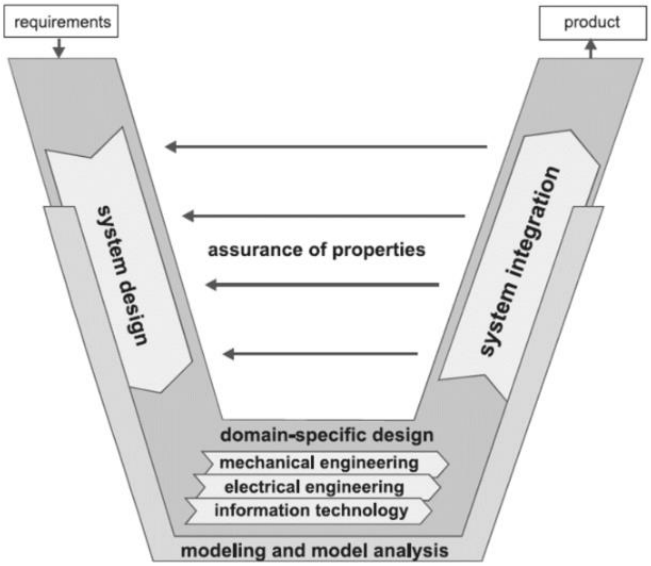


Figure 4.1-3: VDI 2206 V-model design methodology [257].

Regarding its applicability to medical device development, **medical device development standards** define the steps to be followed for the life-cycle development of a medical product. Although they do not prescribe a specific framework, they **describe their processes on V-based methodologies** as examples. IEC 60601-1, the basic standard for guaranteeing medical devices' functional safety, is an example of a V-model for developing a Programmable Electrical Medical System (PEMS). Likewise, this life cycle is in line with the one described by IEC 62304 for the software development of medical devices. Therefore, it can be concluded that the most **direct way to comply with the basic regulations applicable to a medical device** is to use the V-model.

However, it is **not appropriate for start-ups** as they usually **do not have consolidated technical teams, product requirements are not defined from the beginning and developments**, due to their nature, present a **high degree of technological uncertainty**.

Methodol. Req. 2 Comply with the order of tasks and processes predefined in the medical device development standards.

Methodol. Req. 3 Define the verification of each implementation phase before the deployment.

4.1.3 Incremental model

The incremental model seeks a **progressive growth of the product's functionality**. The product will gradually grow in each iteration until it achieves the functionality the end customer requires [258]. It is also known as the Successive version model.

The main characteristic of this methodology is that the **tasks are divided into iterations** to achieve the desired functionality [259]. The iterations of this model are **not independent**; they are developments linked to each other. The increments are usually small, allowing easy management of the tasks. Unlike in Waterfall and V-model, during the early stages of development, the client **can have partial deliverables** enabling them to evaluate how the investment is being materialised. The Incremental model and an example of its outputs are depicted in Figure 4.1-4.

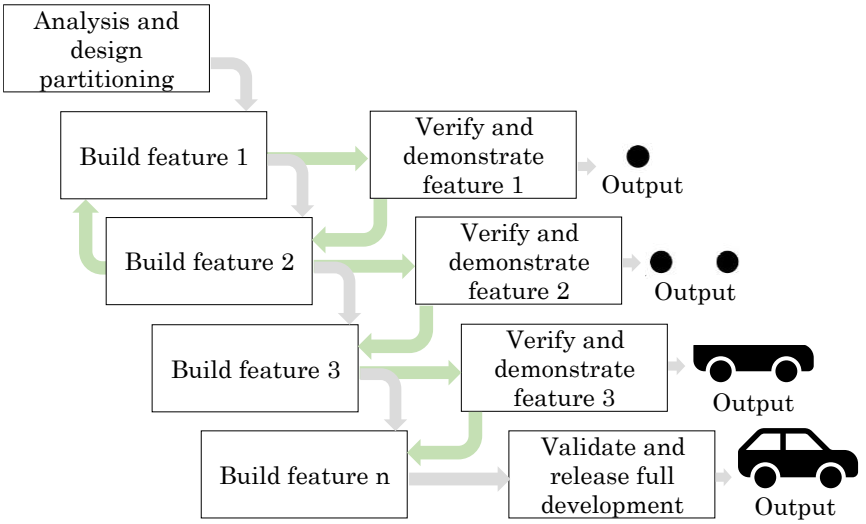


Figure 4.1-4: Incremental model and its outputs.

Unlike sequential and linear models, this model **evidences potential errors or deviations at the early stages** of development. It also has a certain **flexibility** that makes **fixing mistakes easier and faster**.

Regarding **project management**, this approach is more **complex** than Waterfall, as it is necessary to have good planning, manage more phases, have greater communication and coordination with the client, and generate and control different versions of the development. Likewise, it is necessary to have a good vision of the final objective of the development, as the initial planning must consider all the stages of the project [260]. All this means that the **development cost is higher** than the Waterfall methodology.

This methodology is an **advantageous alternative for medical** product development **where the overall requirements are precise**. It is also suitable when there **is much technical uncertainty,** or the **technical team is not highly qualified**. The development can be planned so that the **prototypes** can be **oriented towards reducing uncertainty**, as in the case of start-up companies that wish to develop their first medical product.

Methodol. Req. 4 Enable incremental development to reduce technical and financial risk once the requirements are established.
Methodol. Req. 5 Detect errors in the early stages of development.

4.1.4 Iterative model

The iterative model is based on the **iteration of several Waterfall cycles**. However, this model does not require the system's general specifications to start with the development; the **requirements of only one of the parts are enough** [261]. It begins with the creation of an initial version of the system. Then, it is evaluated, and a new iteration is performed to improve and add functionality to the system. The inputs and outputs of the model are presented in Figure 4.1-5.

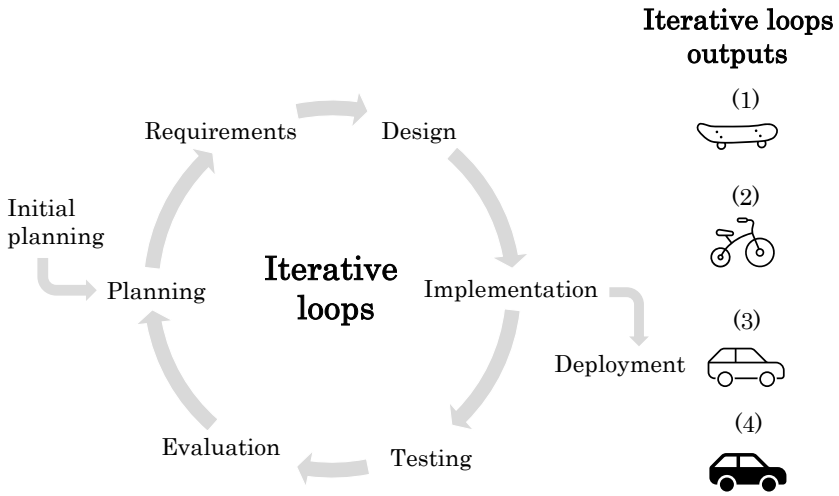


Figure 4.1-5: Iterative model and its four outputs identified as (1)-(4).

This **customer-oriented model** focuses on obtaining user feedback in the early stages of development. For effective feedback, the model requires a good communication plan. User feedback will be the basis of the success of this methodology [262], so it is necessary to involve the client as soon as possible. Therefore, it **allows error detection in early phases**, preventing errors from progressing between phases.

This model is oriented towards **effective risk management** as it reduces risk by providing for constant review and redesign of the product [263]. However, it can be **dangerous for use in projects requiring security mechanisms or with a high degree of risk**, as not all the system requirements are included at the start of development.

On the other hand, the lack of planning and clear requirements from the beginning of the project makes the architecture unclear and the range of possibilities too vast. All this can result in a project deviating in terms of time and cost.

Start-up companies often adopt this methodology to launch their solutions to the market as it allows them to have **a complete system solution at the end of each iteration** [264].

However, its **use** for **medical products** is **not advisable**. Not considering all the safety requirements at the beginning of the development can lead to a **system architecture that does not meet**

the final needs, deviating the project from its targeted lead times and cost. The development of medical products involves the generation of documentation, which further increases as the product evolves. This approach can be interesting for developing functionalities not an essential part of the medical device and where end-user feedback is important, e.g., for designing graphical user interfaces.

Methodol. Req. 6 Obtain client feedback in the early stages of development.
Methodol. Req. 7 Develop iteratively during the initial phases of the project when requirements are not clearly defined.

4.1.5 Spiral model

The Spiral model was developed by Barry W. Boehm in 1986. It is a combination of the Waterfall model and the Iterative model [265]. Spiral is based on a **series of iterations**. After the execution of each of them, **a prototype is available to evaluate product risks and suitability**.

This model is **focused on reducing development risks**. As presented in Figure 4.1-6, typically, it is divided into 4 phases: determination of objectives and identification of alternative solutions, risk identification and analysis, development and testing, and planning of the next iteration [266].

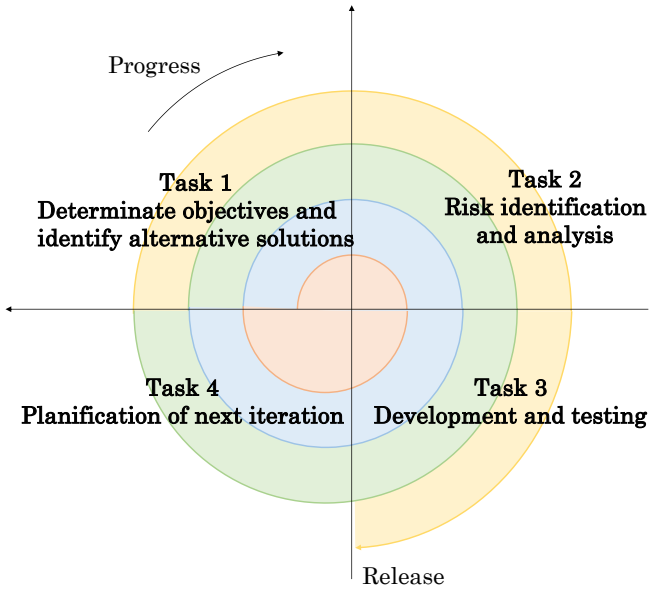


Figure 4.1-6: Spiral model.

Completing a task is required to move on to the next one making the **execution linear and sequential**. At the end of each phase, the necessary deliverables are generated, facilitating the transition between phases. In addition, it is **focused on receiving feedback from the client** in the early stages of the design process. This method requires a continuous dialogue with the client.

The Iterative and Spiral models have similar applications, but in this case, this model places particular emphasis on risk analysis [267]. As discussed above, effective medical device development relies on good risk management. Therefore, this model is more appropriate than the Iterative model for medical device development.

Despite this, it presents the same problem as the Iterative. On the one hand, the **lack of precise specifications can cause development deviation in terms of time and cost**. On the other hand, **due to the sector's requirements, releasing partial product versions to the market could be challenging as the cost can be very high**.

Methodol. Req. 8 Consider risk management in the different iterations of the methodology.

4.1.6 Prototype model

The Prototype model is based on the **development** of systems called **prototypes**. It is an **iterative process of trial and error**. Based on an initial specification, a prototype is developed and evaluated before an improved prototype is developed [268]. The model, as shown in Figure 4.1-7, is composed of six phases [269]:

- Initial requirements: The process starts with the analysis of the requirements.
- Design: A first design of the system is made; this design is usually simple and gives the user a first idea of the final development.
- Build prototype: The system solution is manufactured and integrated.
- Customer evaluation: The client can evaluate the development once the physical prototype is available.
- Refining prototype: Feedback is used to refine the development process after client evaluation.
- Implementation of product and maintenance: The concept is taken to the product after the final prototyping. This phase also includes the maintenance of the product.

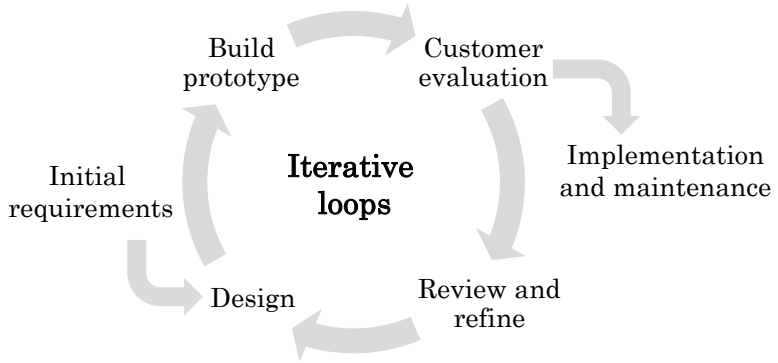


Figure 4.1-7: Prototype model.

This model requires a high degree of customer involvement. Therefore, it is possible to identify whether the development satisfies the user's expectations in the early stages. Also, **errors or deviations can be identified early** and can help reduce cost and time compared to the Waterfall model. On the other hand, as with

Iterative development, sometimes the number of iterations can increase dramatically, leading to higher costs and delays.

The fact that there are **several iterations can result in low-quality documentation**; this is a **potential problem in medical device development**, as documentation is a crucial part of the product certification process. Often, the generation of functional but simplified prototypes can confuse the client as they can think that their development is almost ready in advance. The prototype model requires good communication with the client so that they understand the current status of the product as well as the remaining phases of the development.

Like the iterative process, it is especially **interesting for the development of graphical user interfaces**. Through prototyping, the user can experience how the interface will look at the beginning of the development and make suggestions for changes. It is also **effective when the technical solution is unclear**; prototyping helps minimise the existing uncertainties, for example, in developing new innovative medical devices.

The Prototyping model has three variants: Rapid Throwaway prototype, Evolutionary prototype and Incremental prototype.

- Rapid Throwaway prototype [270]:

This variant tries to develop prototypes quickly. It is enough to have preliminary development requirements as it is oriented to have prototypes for later evaluation. User feedback is what helps to establish the final requirements. In this method, the prototypes are discarded and are not part of the final solution; they are simply a way to define the final requirements.

- Evolutionary prototype [271]:

In this case, incremental development is carried out. There are requirements for incremental prototyping, and the customer's feedback is used to adjust the development. This model is oriented to be used in complex projects where it is necessary to consolidate and verify the product by parts.

- Incremental prototype [272]:

The final development is split into parts. Small prototypes are developed and can then be combined to produce the product. Developments can be carried out in parallel, which reduces development time.

Methodol. Req. 9 Develop rapid prototypes that help define the concept or clarify technical uncertainties.

Methodol. Req. 10 Parallelise tasks to minimise development time.

4.2 Agile Methodologies

The origin of Agile methodologies dates to 1968. This year, a name was given to the cost overruns and quality deficiencies suffered in software developments [273]. At that time, it began to be realised that the initial requirements and the environment were continuously changing. Since then, new methodologies have appeared to **work in dynamic and changing environments, enhancing the product's final value**. Likewise, it is assumed that **people are the core of the projects**; they must be involved in project execution to deal with unexpected events.

Therefore, Agile methodologies were developed to address the massive bureaucracy that traditional methods have to deal with. Agile methodologies **are not oriented towards documentation or development planning**. The fact of not dedicating a huge effort to planning makes these methodologies **easily adaptable to changes**. Also, these methods **are people-oriented and not process-oriented** as they are not focused on developing processes that will work for the team; their role is to support the development team [274].

In 2001, several experts in Agile design created the Agile Alliance, a non-profit organisation to promote Agile development and support organisations that wanted to implement these methodologies [275]. As a first step, this Alliance drafted the Agile Manifesto. This manifesto is defined to promote better ways of developing software. Although Agile was created for software development, the philosophy promoted by this manifesto can be extended to any other product. To this end, it defines 4 core values and 12 principles based

on this philosophy [276]. Core values represent the items that add the most value:

- **Individuals and interactions** over processes and tools.
- **Working software** over comprehensive documentation.
- **Customer collaboration** over contract negotiation.
- **Responding to change** over following a plane.

Regarding the core values, the 17 manifesto authors clarify that, while they recognise the importance of the elements on the right, they place more value on those on the left.

The following are the 12 principles of Agile:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable "software".
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for a shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to do the job.
- A face-to-face conversation is the most efficient and effective method of conveying information to and within a development team.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity – the art of maximizing the work not done – is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.

- The team regularly reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

4.2.1 Key Agile concepts

In Agile development methodologies, some concepts are considered essential. The main terms are briefly detailed to understand Agile frameworks better[277]:

- Incremental and Iterative Development:

Most agile methodologies are based on incremental and iterative development strategies; this means that each product development iteration is usable and new functionality is added for the user in each iteration. They also allow reviewing and repeating activities.

- Backlog:

It is a list of developments (new, changes, corrections, etc.) to be implemented. The product owner will be responsible for organising this list and prioritising the items to be included. This list can have many formats, one of which is the User Stories.

- User Stories:

It is the list of requirements of the product. To define them, questions such as What? How? For what? are usually answered. Likewise, the acceptance criteria are also considered. In addition, user stories must follow the INVEST stand: Independent, Negotiable, Valuable, Estimable, Small and Stable.

- Daily Meeting:

The team holds a daily meeting to identify problems and establish corrective measures. To do this, during the meeting, each member answers the questions: What did I do yesterday? Have I completed the task? What am I going to do today? Do I see impediments to meeting the objectives?

- Milestone Retrospective:

From time to time, or at the end of the project, all team members spend a day analysing the most significant events during the project's development.

Methodol. Req. 11 Ease the introduction of changes.
Methodol. Req. 12 Clearly define the acceptance criteria of each task or iteration.
Methodol. Req. 13 Review and reflect on the progress of iterations.
Methodol. Req. 14 Promote motivation and involvement of the team, and ask them for feedback on the project's overall implementation.

4.2.2 Scrum framework

Scrum is one of the **most extended Agile frameworks**. It originates from a study of new development processes used for successful products in Japan and the United States [278]. These developments were based on generic and innovative requirements and had a very short time to market. When analysing the working patterns of the team, it was observed that they were very similar. This study compared how they worked with rugby players and their scrum training. In 1995, Jeff Sutherland officially defined the Scrum process during the Object Oriented Programming, Systems, Language and Applications (OPPSLA) conference [279].

The Scrum methodology focuses on **experience-based learning**, **self-organisation to address problems and thinking about wins and losses**. Although it is considered an Agile project management tool, this framework **includes meetings and utilities that help teams to structure and manage work**. The main elements of Scrum are [280]:

- Backlog management:

The organisation of activities by the product owner is required. The term Backlog is defined in section 4.2.1.

- Planning:

The planning phase contemplates organising the items to be addressed during the different iterations (Sprint).

- Sprint:

It is the period when the team works together to achieve an increase in development.

- Daily Scrum meeting:

It is defined in section 4.2.1 as Daily Meeting. **Task tracking is done through a dashboard** where the team has a visual overview of the work. This dashboard presents the list of user stories and the status of different subtasks classified as To Do, In Progress and Done. An example of a dashboard is illustrated in Figure 4.2-1.

- Sprint review:

At the end or during the execution of the sprint, the team meets to review the achieved increment.

- Sprint retrospective:

The team meets at the sprint's end to assess and share what went right and wrong during the iteration. **Sprint review and retrospective are based on the Milestone Retrospective** concept defined in section 4.2.1.

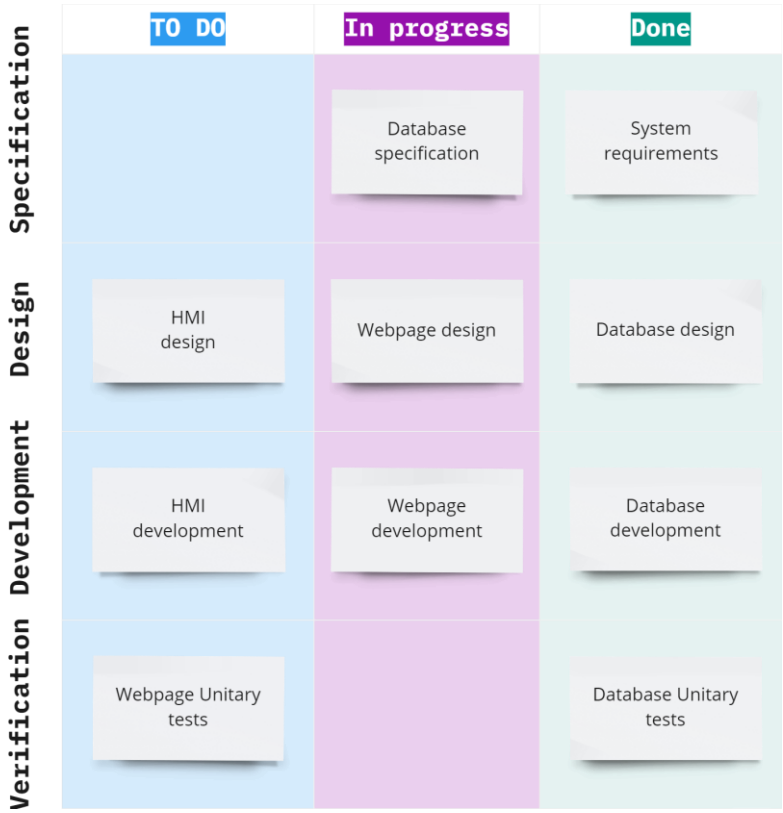


Figure 4.2-1: Example of a Scrum board.

Scrum has **clearly defined roles and responsibilities**. The team usually has 5-15 members, including the Product Owner, the Scrum Master and the development team [281]. The Product Owner optimises and maximises the product's value, managing the value flow through the Product Backlog. Likewise, it is the link between the team and its stakeholders. In each sprint, the Product Owner must identify and mark the value objectives to be achieved.

The Scrum Master is responsible for ensuring the Scrum is carried out correctly. Therefore, it is responsible for managing the Scrum process and the problems that may arise. The development team usually consists of several professionals who are involved in the development of the product. The working team creates the increment for each sprint. In Figure 4.2-2, the Scrum workflow is presented.

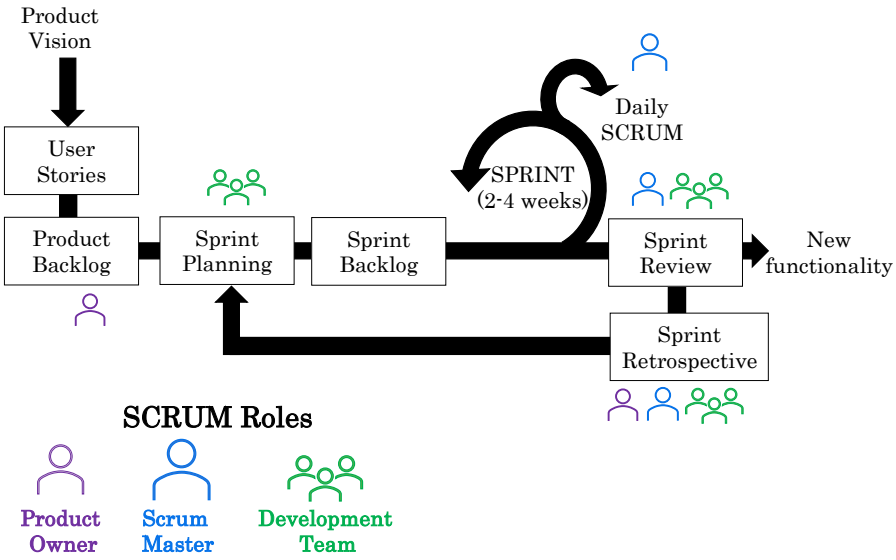


Figure 4.2-2: Scrum workflow.

Agile methodologies, and in particular Scrum, can be **beneficial for start-ups**. These methodologies offer **processes that can answer speedily to customer needs**; this is what these types of companies are looking for from the beginning. It also **helps teams to self-manage to create simplified prototypes**.

On the other hand, Scrum **requires experienced managers to apply the method** to be effective. Also, this methodology **is not easy to integrate with the classical project management approach**, typically used for medical device development. This characteristic may lead to this methodology **not being entirely suitable for developing medical devices**.

Methodol. Req. 15 Use a tool that allows the team always to have a clear and complete overview of the project's status.
Methodol. Req. 16 Define short iterations (Sprints) that deliver continuous value to the client.
Methodol. Req. 17 Involve the team in the planning.
Methodol. Req. 18 Define clear roles.

4.2.3 Kanban framework

Kanban is an **Agile product development framework**. It was defined in 1940 by the Toyota engineer Taiichi Ohno. He developed a system

to control the supply chain stock [282]. This framework is **based on the philosophy of continuous improvement, ensuring quality, waste reduction and flexibility**. Its objective is to **manage development flows visually**. To do so, it relies on dashboards [283].

In 2007, Microsoft employee David J. Anderson formulated the four principles and six basic practices of Kanban [284]. As for the principles of Kanban, they are aligned with the Agile manifesto:

- Start with what you do now:

Kanban **requires no configuration and can be applied to existing workflows**. This framework acts as a value enhancer and a tool to address problems that block processes and results.

- Agree to pursue incremental, evolutionary change:

Kanban recognises that implementing small changes presents less environmental resistance. Therefore, **it relies on small incremental and evolutionary changes**.

- Respect the current process, roles, responsibilities and titles:

This framework does not prohibit or prescribe change as it recognises the current state of roles, titles, responsibilities, etc. The aim is to build **support for people** to show interest **in established practices and gradually bring change**.

- Encourage acts of leadership at all levels in your organization:

The mindset of continuous improvement at all levels is discussed, as it is considered that it cannot be an exclusive management-level activity.

The **Kanban framework allows work to start when there is a demand for deliverables**. In contrast, Scrum has a time-based approach (Sprints). The 6 Kanban practices are:

- Visualize the work, workflow and business risks:

It is necessary to identify which processes and phases the product development should follow, from conception to release. This process is visualised using dashboards in a similar way to Scrum. Usually, in its simplest form, the dashboard includes the following columns: To Do, In Progress and Done.

Also, the dashboard should present the work by representing each task on a card, the workflow, which the different columns and business risk will represent by including colour codes, tags or labelling.

- Limit Work In Progress (WIP):

The idea is to limit the workload on the board to what the team can manage. New work is included on the dashboard only when there is the capacity to manage it.

- Manage flow:

The project manager will be in charge of moving the different cards on the board.

- Make policies explicit:

Decisions should be objective, and the team should have clear and documented rules.

- Implement feedback loops:

In Kanban, feedback is collected at different project stages, follow-up meetings, handovers, etc. The project manager establishes the frequency and form.

- Improve collaboratively and evolve experimentally:

Kanban recognises that continuous improvement and sustainable change are only feasible through a shared vision.

In terms of roles, **two leading roles are defined**: Service Request Manager and Service Delivery Manager [285]. The Service Request Manager is responsible for identifying customer expectations and needs. It is also responsible for selecting and prioritising the work

items. The Service Delivery Manager is responsible for the workflow and must facilitate the Kanban process.

Kanban offers significant **advantages for start-ups**. It can **manage a continuous flow of work input** and supports it anytime, even during a running task. This approach is more **appropriate for team members with work inputs that vary in scope and priority**. It also supports maintenance and support tasks.

On the other hand, Kanban **is not a very structured** method, so it can be **more challenging to implement this framework in a company where the employees are not very experienced**. Moreover, it is u**nsuitable for projects requiring high predictability and fully defined planning**.

Methodol. Req. 19 Identify the category, risk and priority of each task.
Methodol. Req. 20 Seek continuous improvement.
Methodol. Req. 21 Do not impose drastic changes that may be difficult to accept.
Methodol. Req. 22 Accept changes during development, at least partly.

4.2.4 Crystal frameworks

Crystal is a family of **frameworks for Agile development** that **focuses on people and their interactions**. This methodology was defined in 1991 by Alistair Cockburn, creator of the Agile Manifesto. Specifically, the first definition was defined for IBM [286]. This methodology **pursues individual improvement to achieve the team's overall progress**. It focuses mainly on people, interaction, community, skills, talent, and communication. It is usually used in software development, although it can be generalized to the development of integral products containing software.

There are different Crystal methodologies depending on the number of people involved in the development, criticality and project priority. Crystal Clear (less than 8 people), Crystal Yellow (10 – 20 people), Crystal Orange (20 – 50 people), and Crystal Red (50 – 100 people) [287]. All these frameworks are based on 7 basic principles [288]:

- Frequent delivery: Generate very frequent functional deliveries.
- Reflective improvement: Reflect on and review the work being done to identify areas for improvement; this is done through team meetings.
- Osmotic communication: Effective and fluid team communication is sought.
- Personal safety: Communication between the different members is encouraged to communicate any disagreement that could arise.
- Focus on work: Priorities must be clear, and the team must have the right time and environment to focus on the work.
- Access to subject matter experts and users: It is intended that the team members have easy access to experts from different fields.
- Technical tooling: It aims to facilitate the integration and verification of the system. This approach is expected to support the team with automated tools that minimise human error.

Crystal also **defines different roles and responsibilities**: the Project Sponsor, User Representative, Lead Designer, Programmer, Coordinator, Business Expert, Technical Writers and Testers [289].

The Project Sponsor is responsible for defining the development budget, giving visibility to the project and helping the team to make business-related decisions. User Representatives must be able to test the final product and therefore dominate the system's full functionality. Lead Designer is responsible for the technical aspects and must have the expertise to evaluate the team's progress. Developers are the ones who carry out the technical work.

The Coordinator or Project Manager must follow-up on the project and consolidate the information generated to present it to the Project Sponsor. The Business Expert is the one who knows the policies of the company and its priorities. Based on these policies, the Business Expert must ensure that the team fulfils these. Finally, there are the Technical Writers and Testers; different team members can assume these roles to generate documentation and verify the project's development.

The process flow, presented in Figure 4.2-3, is similar to Scrum. It starts with an episode. Episodes are small developments that are the basis for different components. These episodes can be joined together to form an Integration. Iterations are released on a daily or weekly basis. Once the iterations are finished, the project is integrated, verified, and delivered [290].

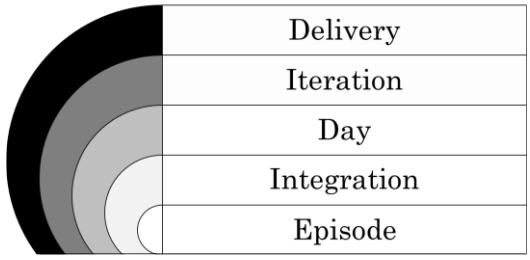


Figure 4.2-3: Crystal workflow.

These frameworks **provide autonomy and freedom for team members to work most efficiently**. They help **reduce management overheads and costs** as the communication between different teams is done directly. Moreover, as they are a family of frameworks, they can be easily migrated between them and adapt to changes in the number of people in each team.

On the other hand, the **lack of processes, rigid planning, clear requirements, and structure makes it unsuitable for inexperienced teams** as it can be complicated and confusing. Also, these frameworks are based on informal and continuous communication between team members, which is **incompatible with teams with remote workers**.

Methodol. Req. 23 Promote communication between the different members of the team.
Methodol. Req. 24 Use technical tooling to assist the team during development and verification.

4.3 Lean Startup Methodology

Lean Startup is a **product design methodology focusing on customer needs**. This method was born as an evolution of the Lean Manufacturing methodology.

The origins of Lean go back to 1890 when Sakichi Toyoda, a textile entrepreneur, created several patents for machines that helped to automate operators' work. Kiichiro Toyoda developed this philosophy by focusing on the collaborative work of machines and humans to add value to production without generating waste. This methodology was called Just In Time (JIT). Eiji Toyoda added value to JIT by defining the Toyota Production System (TPS), which aimed to produce only on demand, thus optimising production times [291].

In 1990, thanks to the article, The Machine that Changed the World [292], published by Wornak, Jones and Roos, this methodology reached the West and the concept of Lean Manufacturing was introduced.

Lean Startup **is a form of Lean defined for new product development** by Eric Riese in 2012 in his book "The Lean Startup" [293]. This methodology emerged to **create successful companies using continuous improvement**. According to Riese, start-ups' success lies in designing the right processes.

Lean Manufacturing is based on creating as efficiently as possible. In contrast, Lean Startup **is based on validating the product concept as early as possible** and **creating the product the customer requires while optimising resources**. This method seeks to **minimise the risk of failure** by preventing start-ups from being unsuccessful due to a lack of customer feedback until the very late stages of product design.

Concept validation is done **by developing a Minimum Viable Product** (MVP). Fran Robinson defined this concept but it was not popularised until Eric Ries introduced it in Lean Startup. According to Ries, **MVP** can be described as *a version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort* [294]. MVP does not have to be a technological development but a way to validate the concept.

Lean Startup is based on five principles [295]:

- Entrepreneurs are Everywhere:

This principle means that there are many entrepreneurs in very different environments, so when creating a new product, you face many competitors.

- Entrepreneurship is Management:

A start-up is an organisation that requires management methods. However, by its nature, it requires tools and methods adapted to its environment and the uncertainty surrounding it.

- Build Measure Learn:

At the beginning of product development, a start-up does not have enough information to create a product that fits the customer's needs, leading to designing a product with no market. Lean Startup proposes a development cycle based on Build, Measure and Learn to minimise the risk of failure.

- Validated Learning:

One of the main objectives of start-ups is to generate sustainable business models over time. Therefore, continuous learning is essential for this type of organisation.

- Innovation accounting:

The Lean Startup method proposes measuring the degree of progress towards the final goal to ensure that the iterations make sense. For this reason, measuring relevant Key Performance Indicators (KPIs) is crucial.

Accordingly, the Lean Startup methodology is **based on three steps executed iteratively** until the development is completed successfully: **Build, Measure and Learn** [296]. The workflow of Lean Startup is presented in Figure 4.3-1.

- Build:

A prototype or a Minimum Viable Product (MVP) must be generated starting from an idea. This phase also includes the establishment of a hypothesis to test the concept.

- Measure:

One of the biggest methodology challenges is measuring customer feedback to allow good decision-making; having the right tools and techniques is necessary.

The author of Lean Startup pointed out that there are vanity and actionable metrics. Vanity metrics make us feel good about ourselves but do not serve us well regarding an honest reflection of business health or as a guideline for our following actions. Actionable metrics are those that provide helpful information to the development team. Actionable metrics refer to quality, not quantity; they take people into account and measure just what is needed.

- Learn:

Adjustments and changes are made to ensure continuous development improvement using the information obtained by the key metrics in this phase.

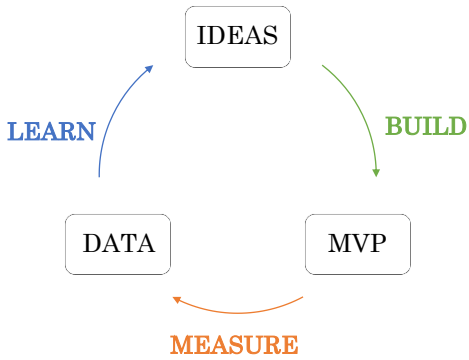


Figure 4.3-1: Lean Startup cycle.

Agile and Lean Startup methodologies aim to satisfy clients through iterative and incremental processes. The main difference is that **while Agile focuses on optimising product development**, **Lean**

Startup defines "Measure" and "Learn" as critical parts of its methodology.

This method will **enable product managers to see results in the short term** as it **optimises** the **investment of money**, reducing the risk to a minimum. In addition, it will **allow having a minimum viable product** that brings value to the specific market segment. On the other hand, this method do**es not consider the administrative and documentation problems** involved **in developing a medical device**.

> **Methodol. Req. 25** Measure development success and progress through KPIs.
> **Methodol. Req. 26** Develop MVP to validate the concept.

4.4 Design Thinking Methodology

Design Thinking is a **human-centred, user-centric way of approaching product design, innovation and problem-solving**. The first reference to the Design Thinking methodology dates back to 1969. In his book "The Sciences of the Artificial" [298], Herbert Alexander Simon, Nobel Prize winner, mentioned this concept. It was not until 2008 that Tim Brown defined the methodology as it is known today. This approach was published in a "Design Thinking" paper in the Harvard Business Review [297].

This methodology is similar to Lean Startup as **it includes prototyping** or MVP development to define and investigate ideas to specify a final solution as soon as possible. The process that follows this methodology contemplates the **non-linear execution of 5 steps: Empathise, Define, Ideate, Prototype and Test**. These phases are executed iteratively until the desired development is achieved [298].

- Empathise:

The first stage is about empathising with the user and understanding their needs and views. Understanding the consumer is considered essential for the success of the product. During this phase, all stakeholders should collect relevant information regarding the idea to be developed. The techniques used can be surveys, interviews, statistics, etc.

- Define:

Once the feedback has been analysed, opportunities for improvement or development can be defined. The team should prioritise them and decide on which ones they start developing.

- Ideate:

It is about coming up with ideas and new alternatives to solve the improvement areas and problems that have been identified. Techniques such as brainstorming may be appropriate for this phase.

- Prototype:

This phase involves developing a prototype that includes some of the ideas raised in the previous stage. This step involves rapid and inexpensive prototyping oriented towards experimentation.

- Test:

In this phase, interviews and tests are conducted to evaluate the proposed solution. The end-user should be involved in the evaluation as their feedback helps to reorient and improve the prototype.

This methodology has several variations adapted to different internationally established companies, such as FJORD (Accenture) [299], Stanford Design School [300], IDEO Design Consulting Company [301], Google Ventures [302], IBM [303], SAP [300] or Apple [304]. SAP's approach to Design Thinking is presented in Figure 4.4-1.

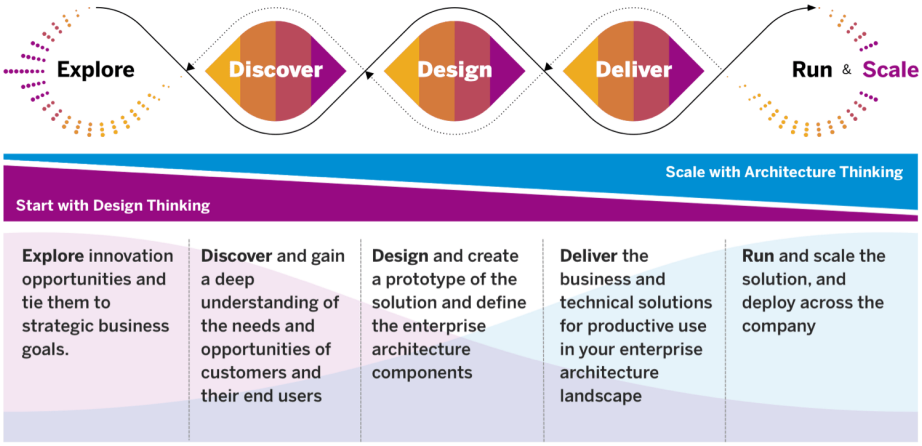


Figure 4.4-1: SAP's approach to Design Thinking [300].

This methodology **eases the adoption of the solution** as it is based on the end customer's needs. It also **fosters the team feeling** of the employees as their ideas are valued in the Ideate phase.

Like Lean Startup, this meth**odology does not contemplate regulatory and documentary requirements**. Therefore, it may be incompatible with the development of medical devices.

Methodol. Req. 27 Define a phase or process to ideate different solutions and alternatives.

4.5 State-Gate® Methodology

State-Gate or Phase-Gate is a **methodology that assists innovation processes while generating new products**. It was presented in 1988 in an article published by Robert G. Cooper in The Journal of Marketing Management [305].

This methodology coordinates the creative process through a structure that facilitates investment decisions. To this end, this methodology is divided into six Stages or phases and five Gates or decision milestones [306], and its workflow is presented in Figure 4.5-1.

Stages are the phases of the new product development process and are divided as follows:

- Stage 0 - Discovery:

In this stage, ideas and opportunities for development are identified. All stakeholders can be involved in this phase. The potential opinions are pre-selected and presented to decide whether they should be taken forward.

- Stage 1 - Scoping:

A preliminary scoping of the idea is prepared to evaluate the product and its market.

- Stage 2 - Business plan concept:

A detailed analysis of each approach is carried out, assessing its technical, commercial, and economic feasibility. This stage includes product definition and analysis, the definition of the business plan, the project plan and the feasibility review.

- Stage 3 - Development:

The prototyping phase of the product and the previously defined plans are executed. Verification and validation plans are also described.

- Stage 4 - Testing and validation:

Product verification and validation are performed. This phase includes all types of testing, such as near testing to identify production errors, field testing or market testing.

- Stage 5 – Launch and implementation:

The last phase opens the gate to manufacturing and market launch. In this phase, the requirements must be aligned for a successful product launch.

Gates are checkpoints where a decision is made whether to continue with the execution of the next phase or **Stage**. These decisions are based on the inputs available at each moment and different metrics that help evaluate the development's state. The decisions that can be taken at each Gate are Go, Kill, Hold and Recycle [307].

- Go: The project is good enough to move on to the next stage.
- Kill: The project is insufficient to move on to the next stage.
- Hold: The project is not good enough to move on to the next stage. It is put on hold to resume in the future, possibly.
- Recycle: The project is good enough to continue developing if some changes are made.

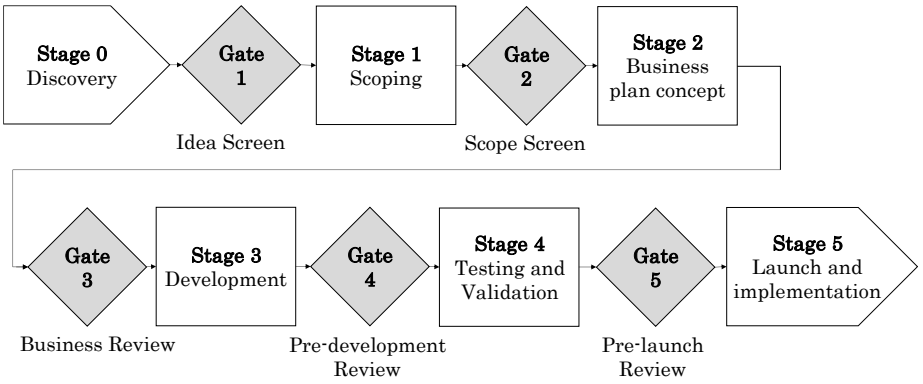


Figure 4.5-1: State-Gate methodology workflow.

The methodology **includes controls related to planning, design and verification**; it also **covers the generation of auditable documentation**. These processes are both required by the different notified bodies responsible for assessing medical devices. In addition, this approach helps **optimise resource use** as the entire context is evaluated at each gate.

On the other hand, this methodology **is not always effective as the high number of gates can become a bureaucratic obstacle** rather than a review opportunity. In addition, this **methodology requires many resources for each idea and, therefore, can lead to a lack of motivation for creating ideas**.

Methodol. Req. 28 Establish project check and evaluation points for the main phases of the project and its associated documentation.

4.6 Hybrid Methodologies

In addition to the traditional methodologies, there is a trend to hybridise several to achieve more effective and efficient methods. The main ones are analysed in the following subsections.

4.6.1 Scrumban methodology

The Scrumban framework was created by **combining the benefits of Scrum and Kanban** [308], [309]. This methodology emerged when different work teams were trying to migrate from one method to the other. During this transition, they identified key aspects that each methodology contributed to effective and efficient product development.

On the one hand, Scrum brings the approach of working in well-defined and time-planned teams to deliver continuous value to end-users. On the other hand, Kanban focuses on process efficiency by ensuring continuous process improvement. Kanban provides a workflow approach using visual dashboards.

Scrumban offers features that are present in the Scrum and Kanban framework. The main ones are detailed below [310]:

- **Work is prioritised** according to complexity and work demand (Scrum).
- There are **different iterations** (sprint) with its **retrospective meeting** (Scrum).
- The **team reviews** and decides when the work is finished (Scrum).
- It works with **limited tasks** to avoid overwork (Kanban).
- **Tasks are identified by cards** (Kanban).
- **Planning** is done by **demand-driven** deliverables (Kanban).
- There is **no product backlog** (Kanban).
- **All team members have the same decision-making opportunities. There is no defined leader** (a new characteristic in Scrumban).
- **Projects do not necessarily need an end** (a new characteristic in Scrumban).

Thanks to these features, **Scrumban saves time, helps manage large projects efficiently, helps the whole team have the same vision,**

provides equity among team members and helps reduce staff stress. On the other hand, this framework is still evolving and **not all best practices are clearly defined**. It also makes **project management difficult** as long as the whole team makes decisions. It is also not feasible to monitor the performance of individual team members.

Scrumban can be helpful when projects include both product development and maintenance phases. It also works best for teams already using Scrum or Kanban, as they already have many of the principles deeply understood. The Scrumban workflow is presented in Figure 4.6-1.

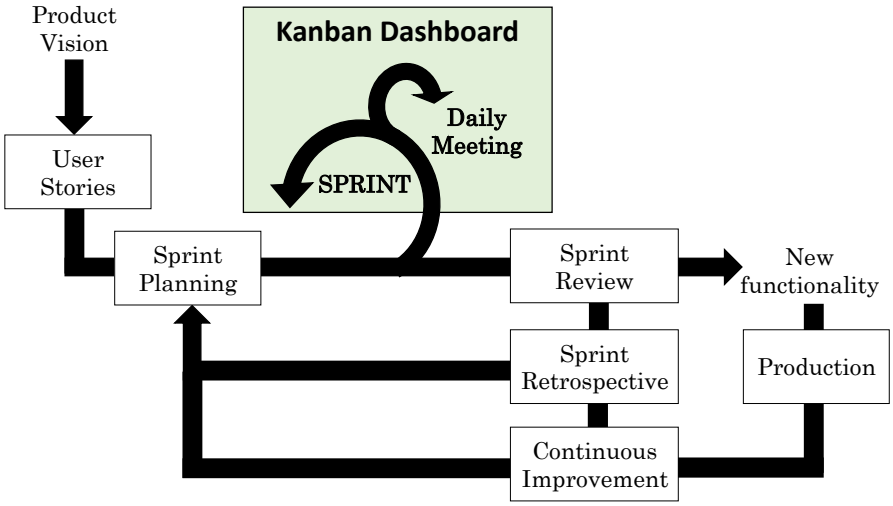


Figure 4.6-1: Scrumban workflow.

4.6.2 Agile-Stage-Gate

Other hybridisations are based on **combining Agile and State-Gate methodologies** [311]. This hybridisation offers new features to make new product development more effective. On the one hand, Stage-Gate provides **a vision for the selection of ideas or projects** to be developed. On the other hand, Agile is more oriented towards project management, offering **techniques and tools for adaptable, time and cost-optimised development** [312].

The hybridisation of the two methods involves agile working modes within each State-Gate stage [313], [314]. Each step has a specific duration defined by the typical duration of the sprint (2-4 weeks). At each sprint, a deliverable is generated, and a review is performed

through the corresponding gate. An example of its workflow is presented in Figure 4.6-2. This hybridisation is more **aware of customer needs**, proactively involves the customer in the process, **reduces iteration time, and is much more productive**.

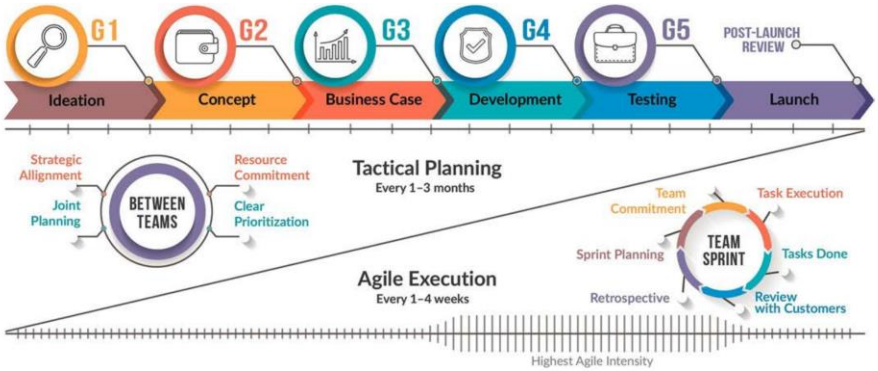


Figure 4.6-2: Agile-Stage-Gate hybrid model [315].

Several manufacturers are already using this type of methodology successfully for the development of all kinds of products. According to [315], [316], Chamberlain, Danfoss, GE, Honeywell, LEGO Group, Honeywell and Tetra Pak use this methodology in 20% of their projects. Specifically, it has been implemented in large projects with higher risk and uncertainty. Before adopting this method, all of these companies were using Stage-Gate. Most of them stated that they migrated to this methodology to resolve internal conflicts, increase efficiency, and answer faster to customers' needs.

Despite being a methodology that can offer many advantages, it is still not a commonly used method, and **some gaps or concepts have yet to be defined** and evolved. For this reason, it is **not easy to implement in start-up companies without experience in design and development methodologies** such as Stage-Gate.

4.6.3 Design-Thinking-Lean Startup-Agile

Another possible hybridisation is the **combination of Design Thinking, Lean Startup and Agile** [317]. Each of the methodologies is optimal in different processes, so when combined, they can increase the value of the method [318].

The process starts with the Design Thinking method, which is suitable for working on the definition of ideas together with the

client. Lean Startup is combined with Agile to validate the hypothesis, define the product's business model and develop the client's solution. This solution is generated iteratively through the execution of different sprints.

Figure 4.6-3 shows a generic process of the methodology. As can be seen, the methodologies are not applied fully; instead, the activities that contribute the most in each phase are selected. Specifically, **Design Thinking is used at the beginning of the development. Lean Startup and Agile are combined to define and develop different concepts**.

This methodology may be **suitable for startups as it provides for resource efficiency**, development **according to customers' specific needs and early detection of potential problems** that may arise. **However, it is unclear how to integrate the documentary requirements** of the medical device regulations into the methodology.

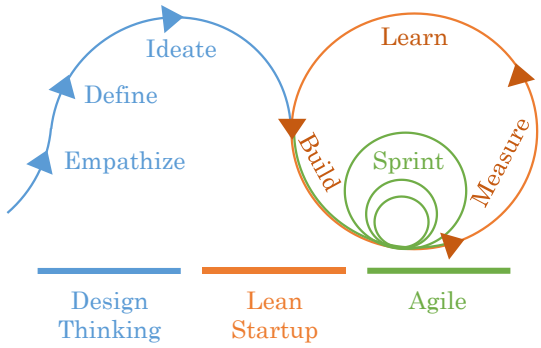


Figure 4.6-3: Generic Design Thinking-Lean Startup-Agile process.

5. Methodology Proposal

A new embedded medical product design and development methodology is presented based on the technical and regulatory requirements and existing product development methodologies. In contrast to other existing methodologies, this approach combines elements of project management, new product development models, embedded systems' technical requirements and medical devices regulatory aspects. The proposal is based on a methodology that supports the development of new products by optimising the cost of development and minimising the associated risk. To this end, a three-stage methodology is proposed. The first stage, Development Feasibility, aims to turn new product ideas into technical solutions, thus minimising the technical and economic risk of the solution. The second phase, Incremental and Iterative Prototyping, is based on the Agile development of functional prototypes iteratively and incrementally. In this phase, the main documentation and procedures of the medical device are established. However, it avoids traditional and rigid medical product development methodologies that do not allow the introduction of changes and obtaining feedback from the client in the early stages of development. Finally, in the third stage, Medical Product Consolidation, the development consolidation is proposed through a V-model that meets all the regulatory requirements for embedded medical device design and development. This methodology is particularly interesting for start-ups due to the type of solution they usually have, innovative ideas but with high technical and economic uncertainties.

5.1 Requirements

This chapter defines the technical, regulatory and methodological requirements to be considered while determining the methodology approach for designing and developing new medical devices based on embedded systems.

5.1.1 Technical requirements

Table 5.1-1 lists the technical requirements or recommendations to be followed when designing a medical device with embedded hardware and software. These requirements will be integrated into the methodology so that technical aspects related to these technologies are considered both in the requirements and design phases.

Table 5.1-1: Technical Requirements

Techn. Req. ID	Definition
1	The hardware platform must include protection mechanisms to guarantee the safety and security of the device.
2	The hardware component o module manufacturer must have ISO 13485 certification.
3	Selected hardware components must be available for the device's lifetime (or alternative components must be available).
4	Use pre-certify components (memories, batteries, HMIs etc.) to ease the certification process of the device.
5	Software resources such as drivers provided by the processor manufacturer must be suitable for certification as part of a medical device.
6	The operating system must provide the required features for the specific application (multi-tasking, real-time extensions, safety and security tools, etc.).
7	The operating system must be certifiable as part of a medical device (consider using pre-certified operating systems).
8	Identify the device's intended use, and consider all use cases to select the most appropriate technology. Typical considerations include use in the professional environment (doctor's surgery, operating theatre, ambulance) and home environment.
9	Components that can cause harm to the patients or operators are considered critical and must be effectively monitored and

Techn. Req. ID	Definition
	controlled to minimise the risk they entail. For example, limiting the maximum current of the device, having redundant temperature sensors in the battery, limiting the use of the device when it is connected to an unstable power source or selecting screens with high luminance, contrast and viewing angle.
10	Use standard communication protocols to ensure interoperability between different devices.
11	Integrate KPIs into critical processes to ease the verification of each component.

5.1.2 Regulatory requirements

Table 5.1-2 lists the regulatory requirements for a medical device's design and development phases. It will be essential that the defined methodology allows compliance with each of the requirements set out here.

Table 5.1-2: Regulatory requirements.

Regul. Des. Req. ID	Definition	Base Standard
1	Identify the classification of the new medical and applicable regulations.	MDR
2	Perform Clinical Evaluation and Clinical Investigation (if required).	MDR
3	Identify the classification of the new medical and applicable regulations.	IVDR
4	Complete Performance Evaluation and Performance Studied (if required).	IVDR
5	Identify device safety classification.	IEC 60601
6	Identify technical requirements to be met (defined in IEC 60601 family).	IEC 60601
7	Perform safety tests to guarantee functional safety.	IEC 60601
8	Perform device safety risk analysis according to ISO 14971 (see Figure 3.6-1).	IEC 60601
9	Identify software safety classification.	IEC 62304
10	Software Development Planning.	IEC 62304
10.1	Plan design and development, verification, risk management, documentation and configuration management (Class A, B, C).	IEC 62304

Regul. Des. Req. ID	Definition	Base Standard
10.2	Document and keep updated software development plan (Class A, B, C).	IEC 62304
10.3	Plan software integration and integration testing, items to control, identification and avoidance of common software defects (Class B, C).	IEC 62304
10.4	Plan software development standards methods and tools (Class C).	IEC 62304
11	Software Requirement Analysis.	IEC 62304
11.1	Document and update software requirements (Class A, B, C).	IEC 62304
11.2	Re-evaluate risk analysis (Class A, B, C)	IEC 62304
11.3	Verify software requirements (Class A, B, C)	IEC 62304
11.4	Include risk control measures in software requirements (Class B, C).	IEC 62304
12	Architectural design.	IEC 62304
12.1	Transform requirements into architecture (contemplate interfaces, functional and performance requirements, SOUP items) (Class B, C).	IEC 62304
12.2	Verify software architecture (Class B, C).	IEC 62304
12.3	Identify segregation necessary for risk control (Class C).	IEC 62304
13	Detailed design.	IEC 62304
13.1	Divide software into software units (Class B, C).	IEC 62304
13.2	Develop a detailed design for each software unit (including interfaces) (Class C).	IEC 62304
13.3	Verify detailed design (Class C).	IEC 62304
14	Software unit implementation and verification.	IEC 62304
14.1	Implement each software unit (Class A, B, C).	IEC 62304
14.2	Establish unit verification process, and define acceptance criteria (Class B, C).	IEC 62304
15	Software integration and integration testing.	IEC 62304
15.1	Integrate and verify software units, and contemplate regression tests (Class B, C).	IEC 62304
15.2	Record integration tests (Class B, C).	IEC 62304
16	Software system testing.	IEC 62304
16.1	Establish tests for software requirements (Class A, B, C).	IEC 62304

Regul. Des. Req. ID	Definition	Base Standard
16.2	Retest after changes (Class A, B, C).	IEC 62304
16.3	Record system testing (Class A, B, C).	IEC 62304
17	Software release.	IEC 62304
17.1	Document released versions (Class A, B, C).	IEC 62304
17.2	Document known residual anomalies (Class A, B, C).	IEC 62304
17.3	Ensure verification is complete (Class A, B, C).	IEC 62304
17.4	Evaluate known residual anomalies (Class B, C).	IEC 62304
18	Perform software risk analysis according to ISO 14971 (see Figure 3.6-1).	IEC 62304
19	Software configuration management: identification of software units and SOUPs.	IEC 62304
20	Risk analysis.	ISO 14971
20.1	Document intended use and reasonably foreseeable misuse.	ISO 14971
20.2	Identification of characteristics related to safety.	ISO 14971
20.3	Identification of hazards and hazardous situations.	ISO 14971
21	Risk evaluation: define risk mitigation strategy for each identified hazardous situation.	ISO 14971
22	Risk Control.	ISO 14971
22.1	Identify risk control measures.	ISO 14971
22.2	Implementation of risk control measures.	ISO 14971
22.3	Residual risk evaluation.	ISO 14971
22.4	Benefit-risk analysis.	ISO 14971
22.5	Review risk arising from risk control measures.	ISO 14971
23	Define use specification (identify characteristics for safety.	IEC 62366
24	Perform a usability risk analysis according to ISO 14971 (see Figure 3.6-1 and Figure 3.7-1).	IEC 62366
25	Perform summative and formative evaluations.	IEC 62366
26	Documentation.	ISO 13485
26.1	Document procedures and records.	ISO 13485
26.2	Review and approval of documentation.	ISO 13485
26.3	Control traceability of documentation.	ISO 13485

Regul. Des. Req. ID	Definition	Base Standard
27	Planning of product realization.	ISO 13485
28	Design and development.	ISO 13485
28.1	Planning: it should cover reviews, verification, validation, transfer, responsibilities, and methods to ensure the traceability of resources.	ISO 13485
28.2	Input and outputs: define requirements (functional, safety and regulatory) and outputs (information for purchasing, production, acceptance criteria, etc.).	ISO 13485
28.3	Systematic reviews: monitor progress and establish corrective actions if necessary.	ISO 13485
28.4	Verification: perform and document verification and check if requirements have been fulfilled.	ISO 13485
28.5	Validation: perform and document validation on a representative product (it contemplates clinical or performance evaluations, EMC...).	ISO 13485
28.6	Transfer to manufacturing: verified development is suitable for manufacturing.	ISO 13485
29	Control the purchases made during the prototyping phase.	ISO 13485
30	Control monitoring and measuring equipment.	ISO 13485
31	Classification of each software item.	IEC 81001-5-1
32	Software Development Planning.	IEC 81001-5-1
32.1	Identify security-related items (IT infrastructure for development, production…).	IEC 81001-5-1
33	Software Requirement Analysis.	IEC 81001-5-1
33.1	Document security requirements.	IEC 81001-5-1
33.2	Review security requirements by experts.	IEC 81001-5-1
34	Architectural design.	IEC 81001-5-1
34.1	Document the secure architectural design, contemplate secure best practices, defence in depth and effective software segregation.	IEC 81001-5-1
34.2	Review the architectural design.	IEC 81001-5-1
35	Detailed design.	IEC 81001-5-1
35.1	Document the secure design.	IEC 81001-5-1
35.2	Identify measures to address the identified threads.	IEC 81001-5-1
35.3	Review the design.	IEC 81001-5-1

Regul. Des. Req. ID	Definition	Base Standard
36	Software unit implementation and verification.	IEC 81001-5-1
36.1	Implement securely (secure coding standards).	IEC 81001-5-1
36.2	Review implementation (security requirements, coding standard, static code analysis, code inspection, traceability, etc.).	IEC 81001-5-1
37	Software integration testing: document and execute.	IEC 81001-5-1
38	Software system testing.	IEC 81001-5-1
38.1	Security requirement testing.	IEC 81001-5-1
38.2	Management of conflicts of interest between testers and developers.	IEC 81001-5-1
39	Software release.	IEC 81001-5-1
39.1	Resolve finding errors.	IEC 81001-5-1
39.2	Document release, including secure operation guidelines.	IEC 81001-5-1
39.3	Integrity verification of scripts, executables, etc.	IEC 81001-5-1
39.4	Verify that security-related issues have been addressed.	IEC 81001-5-1
40	Perform security risk analysis according to ISO 14971 (see Figure 3.6-1).	IEC 81001-5-1
41	Software configuration management: identification of software units and SOUPs.	IEC 81001-5-1

5.1.3 Methodology requirements

Table 5.1-3 shows the most interesting characteristics extracted from the analysed methodologies. These characteristics will be taken as a requirement to define the proposed methodology.

Table 5.1-3: Methodology requirements.

Methodol. Req. ID	Definition	Base methodology
1	Divide the project into sub-tasks to allow the team to focus on a particular development and manage the development better.	Waterfall
2	Comply with the order of tasks and processes predefined in the medical device development standards.	V-model

Methodol. Req. ID	Definition	Base methodology
3	Define the verification of each implementation phase before the deployment.	V-model
4	Enable incremental development to reduce technical and financial risk once the requirements are established.	Incremental
5	Detect errors in the early stages of development.	Incremental
6	Obtain client feedback in the early stages of development.	Iterative
7	Develop iteratively during the initial phases of the project when requirements are not clearly defined.	Iterative
8	Consider risk management in the different iterations of the methodology.	Spiral
9	Develop rapid prototypes that help define the concept or clarify technical uncertainties.	Prototype
10	Parallelise tasks to minimise development time.	Prototype
11	Ease the introduction of changes.	Agile
12	Clearly define the acceptance criteria of each task or iteration.	Agile
13	Review and reflect on the progress of iterations.	Agile
14	Promote motivation and involvement of the team, and ask them for feedback on the project's overall implementation.	Agile
15	Use a tool that allows the team to have a clear and complete overview of the project's status at all times.	Scrum
16	Define short iterations (Sprints) that deliver continuous value to the client.	Scrum
17	Involve the team in the planning.	Scrum
18	Define clear roles.	Scrum
19	Identify the category, risk, and priority of each task.	Kanban
20	Seek continuous improvement.	Kanban
21	Do not impose drastic changes that may be difficult to accept.	Kanban
22	Accept changes during development, at least partly.	Kanban
23	Promote communication between the different members of the team.	Crystal

Methodol. Req. ID	Definition	Base methodology
24	Use technical tooling to assist the team during development and verification.	Crystal
25	Measure development success and progress through KPIs.	Lean Startup
26	Develop MVP to validate the concept.	Lean Startup
27	Define a phase or process to ideate different solutions and alternatives.	Design Thinking
28	Establish project check and evaluation points for the main phases of the project and its associated documentation.	State-Gate

5.2 Methodology Proposal

This chapter presents the proposed methodology approach to address an embedded medical device's design and development stages. To do so, it presents the audience and context of the methodology and the different phases that constitute it. Then, each stage is detailed, focusing on deliverables generated in each design and development step.

5.2.1 Methodology audience

This methodology is intended to address the needs faced by medical product design and development teams. It is expected to provide efficient **uncertainty management** and **cost control** during all design and development stages.

It **covers the regulatory requirements** for these types of devices and can therefore be **applied to any embedded medical device development team**. This medical device design and development methodology can be **used by well-established companies** with an internal design team, research centres that design part or the whole of medical devices for other entities, or even **start-ups with no experience** in the sector.

Despite the aforementioned, as discussed in the previous chapters, start-up companies face added difficulties during the design of medical devices because of their nature and inexperience in the

sector. Therefore, this methodology focuses on addressing the needs of these companies.

This methodology **focuses on the development of new innovative products**. That is, it is envisaged that starting from a preliminary creative idea, this will end up in an embedded medical product. In these cases, the manufacturer lacks precise requirements and the technical solution to be implemented. During the initial phases of this type of project, there is a high level of uncertainty that usually decreases as technological development progresses. The scope of the methodology is presented in Figure 5.2-1. Therefore, this methodology **seeks to manage uncertainty and minimise investment during these phases**. Nonetheless, this methodology is defined in **different stages that can be partially executed**. It can also be applied in those developments in which the requirements and the technical solution are precise.

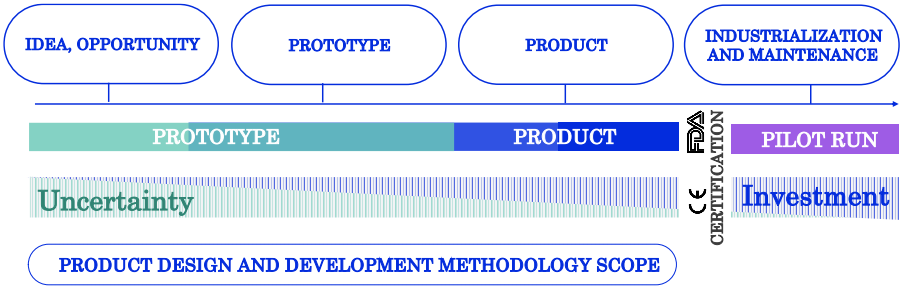


Figure 5.2-1: Scope of the proposed methodology.

Therefore, it is **defined to design the typical embedded systems** that include the following technological blocks: embedded processors, storage units, monitoring or measurement systems, wired or battery power systems, user interfaces and communications interfaces. That is because only the standards that regulate this type of device have been integrated, these being: IEC 60601 (Basic safety and essential performance), IEC 62304 (Medical device software), ISO 14971 (Medical device risk management), IEC 62366 (Medical device usability), ISO 13485 (Medical device quality management systems), IEC 81001-5-1 (Medical device cybersecurity) and several battery regulations.

In case other applicable standards are identified for a design, it will be necessary to review whether this methodology needs to be modified or whether it is directly applicable. It is essential to

mention that the **reviewed regulations are the core standards for any medical device design, so any new standard should be easily integrated**.

5.2.2 Methodology stages

The proposed methodology is based on 3 phases: proof of concept, incremental and iterative prototyping, and medical product consolidation.

- Phase 1 - Development feasibility: In the first phase, idea conception, the methodology aims to create a stage in which different ideas, solutions, and alternatives for developing the concept are identified and validated. This phase seeks to minimise the principal risks and uncertainties present in the development. To this end, **this phase involves analysing and validating the main concepts by developing prototypes with minimum functionality**.
- Phase 2 - Incremental and iterative prototyping: In this phase, the aim is to develop the first integrated prototype, that is, **the first solution of the future medical product.** The development starts with a prototype with limited functionality. Then different development loops of this phase are executed to complete and refine the prototype's functionality.
- Phase 3 - Medical product consolidation: **The prototype must evolve into a product** at this stage. It is crucially important that the development is validated to comply with the defined regulation. A significant part of the **activity will be devoted to verifying, validating, and documenting the product**. For this phase, the methodology proposes to follow a V-model development strategy.

After the execution of each phase, there will be checkpoints or gates that guarantee the correct and expected progress of the project. At this point, both the client and the development team will evaluate the progress and suitability of the solution. The **checkpoints** will also be used **to decide whether or not to continue with the proposed development**. Figure 5.2-2 presents the phases within the proposed methodology's scope diagram.

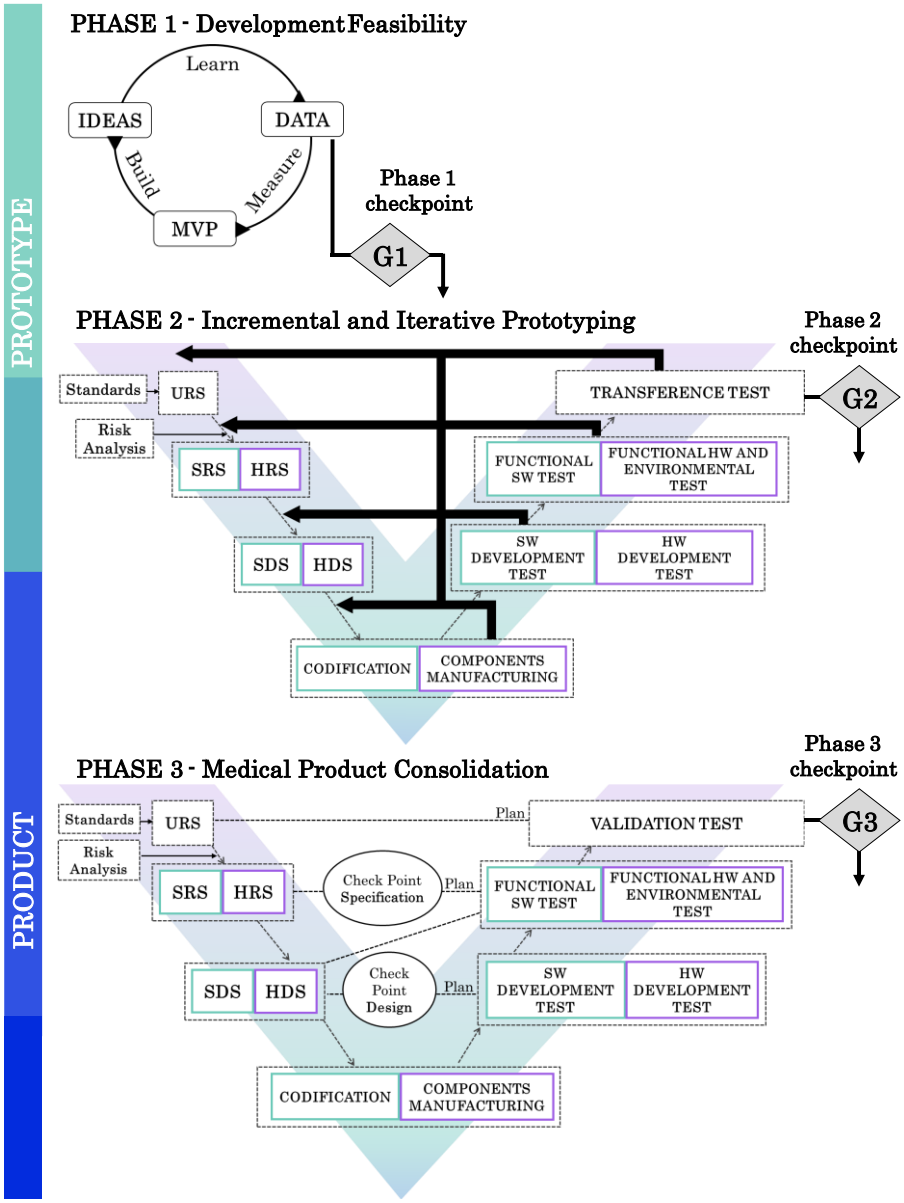


Figure 5.2-2: The stages of the proposed methodology.

5.3 Phase 1 - Development Feasibility

This phase investigates and defines the idea or concept to be developed. Several concept tests are carried out to have a first approach to the solution (Methodol. Req. 27). As mentioned above, the intended use and risk analysis associated with a medical device are crucial to determining its applicable regulations.

Therefore, during this phase, tasks and approaches needed to **define the intended use** and **minimise uncertainties or potential risks** must be executed (Methodol. Req. 8). Usually, these tests are related to the core characteristics of the future medical device. Without the successful performance of such features, the device could not usually be launched to the market.

Figure 5.3-1 presents this phase's key characteristics, inputs and outputs.

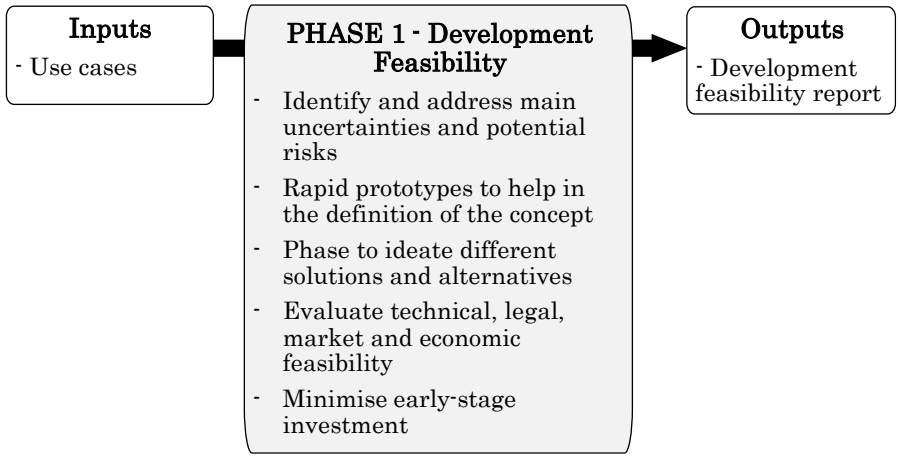


Figure 5.3-1: Development Feasibility phase: characteristics, inputs and outputs.

As a starting point, it is common to have a general idea of the development, usually related to the core of the future medical device. Usually, the main use case that could be potentially useful for the medical sector has been identified. However, there is no specific idea of how to solve it technically. At this phase, there is a slight idea that using an embedded system including different peripherals would be possible to solve the measurement or monitoring principle; however, its technical feasibility is not guaranteed. Hence, during

this stage, parameters to be monitored or measured for diagnosing a physical condition or disease must be clarified or defined.

Hence, this phase should include those tests or proofs of concept needed to overcome the main uncertainties of the project. Particularly those that could make the future product not feasible.

Therefore, the **generated prototypes** will be used for two purposes. On the one hand, prototyping will **minimise technical uncertainty** faced by the development team (Methodol. Req. 9). Prototypes are usually used to clarify parameters such as processing capacity, hardware performance, sensing stages' reliability, maximum latencies or touch panel precision.

On the other hand, the **prototypes can be used to allow the final client to make high-level decisions on the concept**. If there is no access to the end client, this role will be taken by a person outside the technical department. This responsibility usually falls on the department or person designated as a product manager.

For example, suppose a customer wants to introduce a barcode reader. In that case, the different typologies will be analysed in this phase, and tests will be conducted to select the one that best meets the client's needs. Suppose a touch panel is to be introduced, but its typology is unclear in this phase. In that case, the client must be provided with different minimum viable prototypes to verify capacitive and resistive panels (Methodol. Req. 26). In this way, feedback from the client can be obtained, and the related technical (Methodol. Req. 6) solution can be defined.

The identified tasks will be listed and specified (Methodol. Req. 1), and a specialist will execute each. In this way, **parallel execution of some tasks** (Methodol. Req. 10). However, the feasibility will be validated as a whole, so the project manager must be responsible for fostering communication among team members (Methodol. Req. 23).

5.3.1 Roles

The working team is usually formed with different profiles coordinating the team, developing the software and hardware tests, developing the measuring principle, monitoring patent compliance and regulatory consulting. Table 5.3-1 lists the most relevant

profiles and tasks they will perform during the project (Methodol. Req. 18).

Table 5.3-1: Roles and description of activities implemented by the participants of the development feasibility phase.

Role	Activities implemented
Project Manager	- Coordination of project activities and economic and technical monitoring of the project. - Responsible for client communication. - Analyse the feasibility of the whole solution and architecture draft. - Identify weaknesses and strengths, and review development risks. - Review and release of deliverables. - Calculation of the unit and development cost of the device.
Software Engineer	- Responsible for executing software-related tests. - Development of required mocks for the performance of the tests. - Identification of critical uncertainties in software development.
Electronic Engineer	- Responsible for executing electronic-related tests. - Identification of main peripherals. - Identification of key uncertainties in electronic development.
Mechanical Engineer	- Responsible for executing mechanical-related tests. - Development of prototypes to overcome the uncertainties related to electromechanical integration. - Identify critical uncertainties in electromechanical integration.
Optical/Chemical/Physical Engineer (The profile will depend on the technology related to the measurement or monitoring core)	- Identify uncertainties related to the measurement or monitoring system. - Analyses, tests and needed calculations for defining measurement or monitoring core.
Intellectual Property Analyst	- Review of whether the concept to be developed can be patented.

Role	Activities implemented
Regulatory Consultant	- Assist the project manager during the identification of regulatory requirements. - Regulatory requirements should be identified by a specialised department or the customer (depending on the case). However, under no circumstances should the technical department take on this role, as the lack of specific knowledge may make the project fail.

5.3.2 Deliverable: Development Feasibility Report

During this phase, the **development feasibility document** will be generated. The purpose of this document is to create enough information to decide if the development is feasible or not. This document should **assist in evaluating the technical, economic and market feasibility**.

Although this methodology only considers the design and development phases, excluding the business or market that the future device may have, it has to generate information for clients or product managers to decide on these aspects. The information the development feasibility report includes is summarized in Table 5.3-2.

Table 5.3-2: Sections and description of the information included in the development feasibility report deliverable.

Section	Description
Use cases	The first approximation of the main use cases should be identified and drafted in as much detail as possible. This information should be input from the client. It should be drafted with the client or product manager if it is not provided directly as a deliverable.
Uncertainties and potential risks	Based on the information provided in the use cases, the potential risks and uncertainties to be overcome must be identified at the beginning of the project. This identification must be made in collaboration with the client or product manager,

Section	Description
	who knows or must define the device's intended use.
Planification and scope	Once the starting point, uncertainties and use cases are known, the scope of the phase (tests to be carried out, prototypes to be generated, etc.) must be defined and planned. Before sharing the planning with the client, it must be reviewed by the development team (Methodol. Req. 17). This information is collected in deliverables. The client must accept it before the execution of the feasibility analysis phase tasks. The client's acceptance and review of this point is critical as it helps ensure a shared development vision (Methodol. Req. 12).
Technical feasibility results	This section will include the results of the tests and the main conclusions on technical feasibility. This information must be clear and precise. The project manager will arrange a feedback meeting with the team to write the conclusions. Considering their point of view, this section will be completed (Methodol. Req. 14).
Legal feasibility	Information regarding conflicts between the proposed solution and an already registered IP should also be included. Although compliance with medical regulations is not the aim of this phase, possible problems in complying with these regulations should be analysed and identified.
Market analysis	The team should search for similar devices already available on the market. The technical solution and its cost will be analysed (if possible). This information will be presented objectively to compare it with the proposed solution.
Economic analysis	A first approximation of the unit cost of the device and the cost of product development (phase 2 and phase 3) should be made.

5.3.3 Development Feasibility Checkpoint

Once the feasibility phase has been executed, the project manager must deliver the associated deliverables and present the most relevant conclusions to the client.

At this point, a **decision must be made whether the development of the first integrated prototype and the future product should start**

(Methodol. Req. 13) (Methodol. Req. 28). This decision usually involves the upper management as they typically have the final control over the activity. Also, for the final approval, the client company or department should have other inputs generated by other departments, such as finance or sales.

5.4 Phase 2 – Incremental and Iterative Prototyping

After clarifying the uncertainties of the project, it is possible to start with the incremental and iterative prototyping phase. This phase **aims to design functional prototypes that answer to use cases and requirements** defined by the client (Methodol. Req. 26). As the **output** of this phase, a **fully functional and integrated prototype** design of the medical device will be available. However, these will be partially verified and documented. Figure 5.4-1 presents the incremental and iterative prototyping phase key characteristics, inputs and outputs.

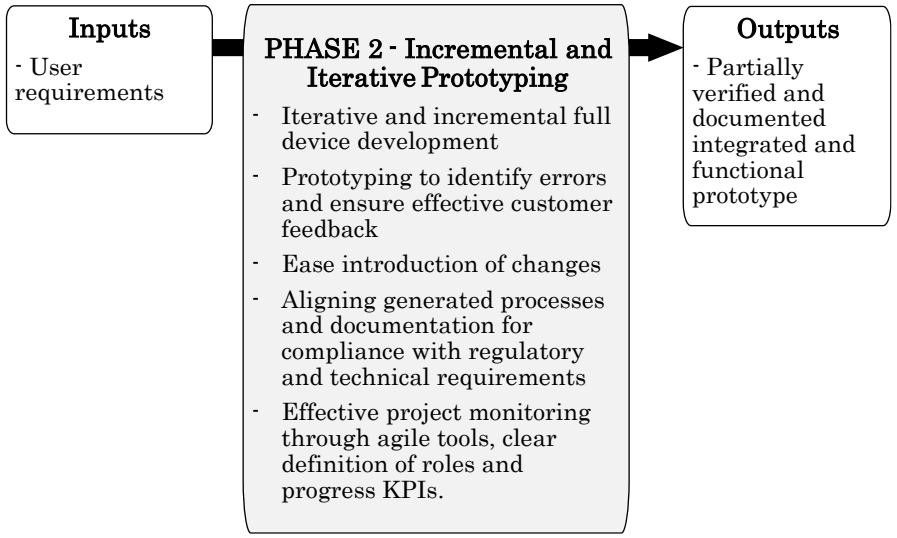


Figure 5.4-1: Incremental and Iterative prototyping phase: key characteristics, inputs and outputs.

To this end, the design and development of the full functionality are proposed using an **Agile methodology based on the iterative** (Methodol. Req. 7) **creation of incremental prototypes**.

Incremental development is considered because it is usual not to have precise requirements at the beginning of this phase. The incremental development will allow the **introduction of requirements and changes in each iteration** and the **reduction of technical and financial risk** (Methodol. Req. 4).

An iterative model is also proposed **allowing clients to assess the suitability of the approach at each iteration** end. The Agile framework of the proposed model for phase 2 is presented in Figure 5.4-2.

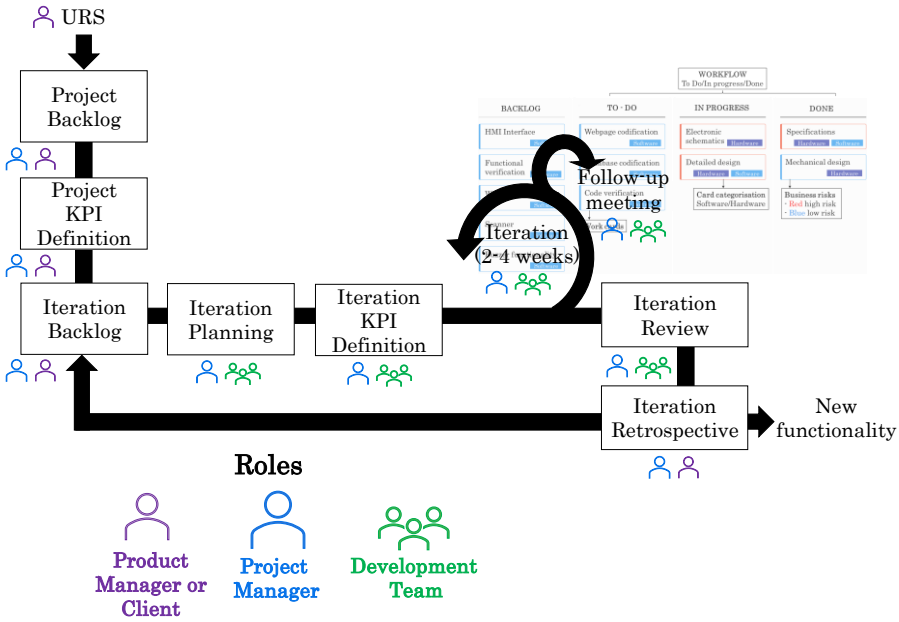


Figure 5.4-2: Proposed Agile methodology framework.

Furthermore, the proposed method is based on some adaptations of the fundamental concepts of agile methodologies:

- Incremental and iterative development:

The methodology proposes incremental and iterative development of the different functionalities of the future product. It begins with those that provide the most outstanding value; they are identified, analysed and developed parallelly by the team (Methodol. Req. 10). Once the generated output is evaluated with the client, iterations are carried out until the desired functionality is completed.

- Roles:

Three leading roles will be defined (Methodol. Req. 18), the product manager or client, the project manager and the development team. The product manager will be responsible for defining the input requirements for this phase. It will also participate in defining project backlog and KPIs, the backlog of the iteration and the retrospective meeting of each iteration.

The project manager will accept the requirements proposed by the product manager and the project backlog, define the iteration backlog and project and iteration KPIs, plan and control the iterations, and coordinate the different development iterations. It will also coordinate the iteration follow-up, review and retrospective meetings.

The development team will undertake the technical activities and participate in planning the iteration, follow-up meetings, definition of the iteration KPIs and the review meeting of each iteration.

- Key Performance Indicators (KPIs):

KPIs will be defined to help objectively measure the project's development and guarantee the project's progress (Methodol. Req. 25). Two groups of KPIs will be defined, project and iteration KPIs. The project KPIs will be established in a meeting between the project manager and the client or product manager.

On the other hand, the iteration KPIs will be parameters that will be measured and defined between the project manager and the team members. The project KPIs will be evaluated in the retrospective meetings between the project manager and the client. On the other hand, the iteration KPIs will be updated during the follow-up meetings and their evaluation during the iteration review meetings.

- Project backlog and user stories:

A list of all the activities to be undertaken throughout the project is available (backlog). These activities will be carried out in several iterations of the proposed model (Methodol. Req. 16). Each action to be executed will be described, prioritised, and categorised based on the functional block to which it belongs.

The information about each activity will be known in this methodology as a user story. The project and product managers will be responsible for completing and updating this information.

- Iteration planning and iteration backlog:

At the beginning of each iteration, the activities to be developed and the available deadlines will be defined. The actions executed during an iteration will be defined as an iteration backlog. The product manager and project manager will decide the backlog of each iteration.

During this meeting, the introduction of possible changes will be assessed (Methodol. Req. 22). If drastic changes are identified, the team will analyse the impact and decide whether it is feasible to undertake them (Methodol. Req. 21). The project manager will allocate the tasks among the team members. Furthermore, all the team will be involved during the planning process (Methodol. Req. 17).

- Follow-up meeting:

Two weekly meetings will be held to analyse the progress of the tasks (Mondays and Thursdays). In these meetings, the team will briefly comment on its development status and, if there are any, will explain its problems. These meetings are intended to let all the team know the development progress and allow the project manager to update iteration KPIs and take corrective actions to ensure that the development is carried out according to the planning. Additionally, these meetings will help to promote communication between the different members of the team (Methodol. Req. 23).

A digital dashboard will be used as a support tool in which the team will update the status of their tasks. The dashboard will present the cards with the prioritised and categorised user stories, the status of each task, dates and responsible (Methodol. Req. 15), (Methodol. Req. 19). Figure 5.4-3 shows the proposed dashboard format. The project manager will be responsible for coordinating this meeting.

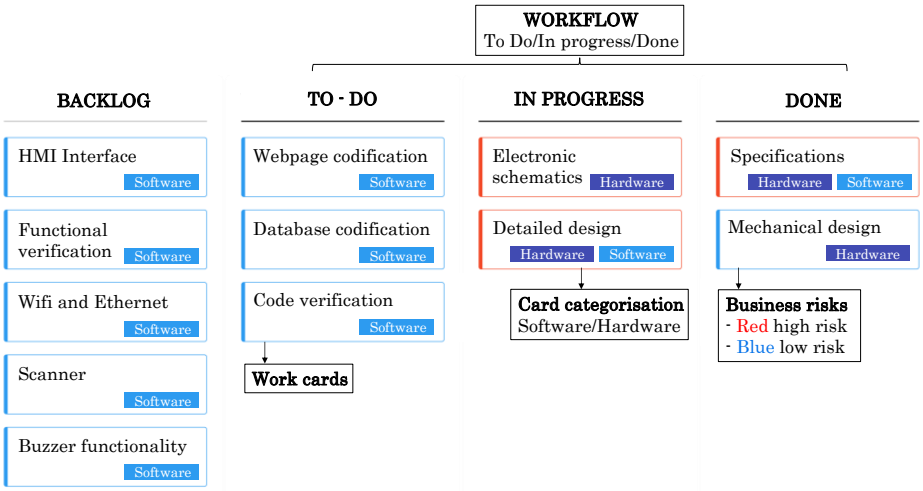


Figure 5.4-3: Proposed Agile methodology dashboard.

- Iteration review:

After each iteration, a meeting will be held to review the completion of the tasks (Methodol. Req. 13). If some activities have not been completed, these should be re-planned later. Also, during this meeting, the execution of the iteration will be evaluated. Feedback from all team members will be collected (Methodol. Req. 14). If necessary, corrective actions will be taken to improve following iterations and guarantee continuous improvement (Methodol. Req. 20). The progress of iteration KPIs will also be reviewed during this meeting. The project manager and the development team will participate in this meeting. However, the project manager will be responsible for coordinating it.

- Iteration Retrospective:

After reviewing the progress of the iteration internally (project manager and team members), the conclusions and actions set will be communicated to the product manager or client. The results will then be reviewed with the product manager, and the progress of the project KPIs will be evaluated.

5.4.1 Overall overview of the proposed model and its deliverables

This section describes the presented model's processes and the model execution flow. Figure 5.4-4 illustrates the proposed model.

Although it is represented in a **V-shape** and not in a circular way, this model considers iterations to change or add new requirements, specifications, design, or implementation.

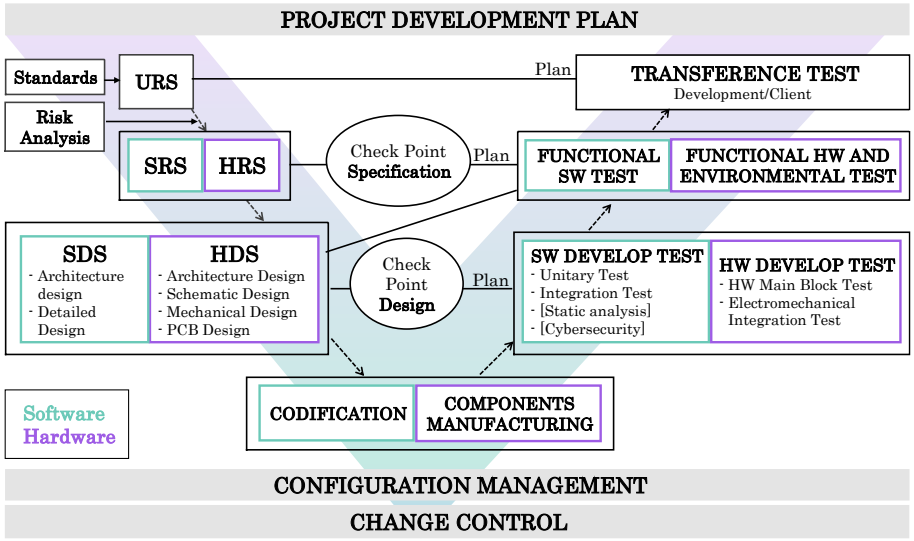


Figure 5.4-4: Proposed V-model for phase 2.

In the case of a change in a user requirement, the entire V-model must be rerun. On the other hand, if the change affects a software or hardware specification, it will be required to execute the specification, design, implementation and verification phases, including the revision of verification plans and their execution. When the change involves a design modification, a review of the development verification plan and execution of the design, implementation and verification phases will be performed. Finally, the implementation and verification phases must be executed if the implementation changes.

This methodology, **through the execution of partial iteration loops, seeks to facilitate the introduction of changes during development and after the product's release** (Methodol. Req. 11). Figure 5.4-5 shows the defined V-model with several iteration loops.

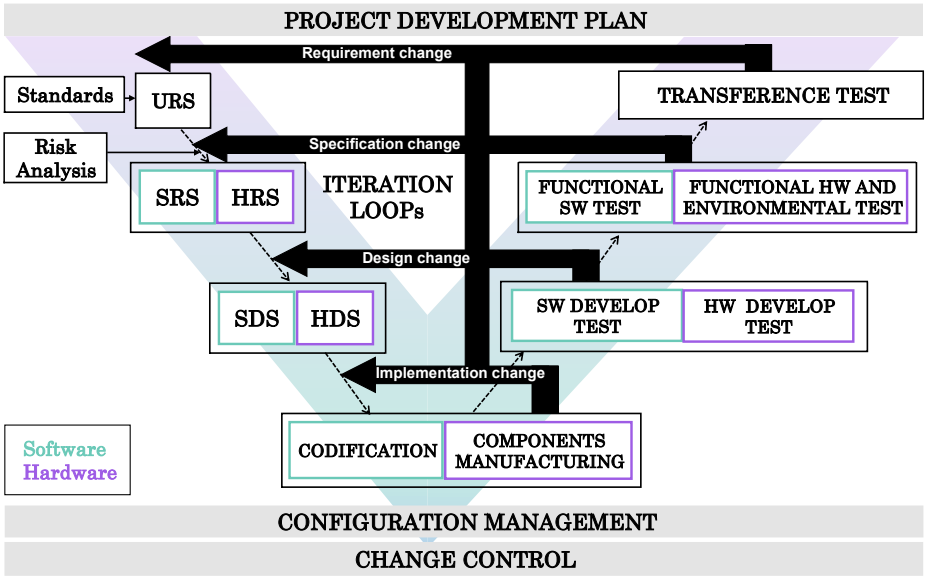


Figure 5.4-5: Simplified model for phase 2, V-model with iterations.

Although the main objective of this phase is not to ensure compliance with the medical regulations, **the proposed model and the generated deliverables are defined to be fully compliant after consolidation and validation of the design and development during phase 3.**

The process starts with the definition of the Project Development Plan. It will define the process to be followed during the execution of the project. Then the Project Transference Plan will be determined by taking the User Requirement Specification (URS) as a starting point. The URSs refer to the definition of functional and regulatory requirements.

The Transference Plan contemplates how the project transfer will be carried out, the acceptance criteria (Methodol. Req. 12) and which tests have not been covered by the tests executed by the development team. The client, product manager, or external team members must perform these tests during phase 3.

Likewise, the transfer document must contemplate when the client has an output of the development to evaluate it and thus continue with the development. This methodology proposes to generate a deliverable for each iteration of the method. This deliverable will

assess the development's progress and clarify technical uncertainties (Methodol. Req. 6), (Methodol. Req. 9).

Then, knowing the URSs and the use cases of the future product, the risks of the product are analysed. The risk analysis and the URSs will be the starting point for defining the project specifications: Software Requirement Specification (SRS) and Hardware Requirement Specification (HRS). Once this definition is available, the Software and Hardware tests for the Functional Verification Plan must be defined. These tests shall verify each of the defined SRS and HRS. A specification control milestone will verify that both the definition of specifications and the definition of the associated tests have been carried out correctly (Methodol. Req. 5).

After the specification phase, the design phase follows. Each HRS and SRS must be translated into a design, Software Design Specification (SDS) and Hardware Design Specification (HDS). Also, the Software and Hardware Development Test Plan that must verify these design specifications is defined (Methodol. Req. 5).

Once the design has been carried out and the associated tests have been defined, a design control milestone must be set to verify that progress is as expected. Once the control milestone has been passed, the software coding and manufacturing of the hardware components can proceed.

After the implementation, the defined verification plans must be executed, first the Development Verification Tests, then the Functional Tests and finally the Transference Tests.

In addition, two transversal processes must also be considered. On the one hand, Configuration Management defines how each project component is formed and tracks its evolution. On the other hand, Change Control manages the modifications performed during the project's design and development phase and after the product's release.

Regarding deliverables, each phase described previously has at least one output that records the activities performed. As the medical device regulation details, **each step of the development process must**

be planned and documented; this implies many plans and records among the documentation defined within this methodology.

The documentation generated in this phase will be the initial version of the final product documentation. Although the documentation effort is minimised in this phase, it should be generated as the development progresses. However, all the information will be consolidated, and the final product documentation will be generated in phase 3.

Due to the large number of documents a medical product development project generates (specifications, designs, verification plans, project development procedures, verification records, etc.), a **master document containing each generated document**, their description and location is proposed [PROJ_MASTER] (Regul. Des. Req. 26), (Regul. Des. Req. 26.1).

Table 5.4-1 shows a suggestion for this document. It also details all the deliverables this design and development procedure generates and a proposal for their location within the project's storage system.

Table 5.4-1: Master document proposal including all project deliverables of phase 2.

Name Document	Description	File Location
Management		
PROJ_OFFER	Project Offer	00 – Management\00 – Offer
PROJ_MINUTES_YYMMDD	Project Meeting Minutes	00 – Management\01 – Meeting Minutes
PROJ_DEVPLAN	Project Development Plan	00 – Management\02 – Development Plan
PROJ_CHKPOINT_SRS	Project SRS Check Point	00 – Management\03 – Check Points
PROJ_CHKPOINT_HRS	Project HRS Check Point	00 – Management\03 – Check Points

Name Document	Description	File Location
PROJ_CHKPOINT_SDS	Project SDS Check Point	00 – Management\03 – Check Points
PROJ_CHKPOINT_HDS	Project HDS Check Point	00 – Management\03 – Check Points
Requirements		
PROJ_URS	Project URS	10 – User Requirements
PROJ_DEV_DEF	Project use cases	10 – User Requirements
PROJ_URS_TT	Project URS and Transference Test Traceability Matrix	10 – User Requirements
Risks		
PROJ_RSK	Project Risk Analysis	20 – Risks
Specifications		
PROJ_SRS	Project SRS	30 – Specification\00 – Software
PROJ_HRS	Project HRS	30 – Specification\01 – Hardware
PROJ_SRS_FT	Project SRS and Functional Test Traceability Matrix	30 – Specification\00 – Software
PROJ_HRS_FT	Project HRS and Functional Test Traceability Matrix	30 – Specification\01 – Hardware
Design		
PROJ_SDS	Project SDS	40 – Design\00 – Software
PROJ_HDS	Project HDS	40 – Design\01 – Hardware
PROJ_SDS_DEVT	Project SDS and Development	40 – Design\00 – Software

Name Document	Description	File Location
	Test Traceability Matrix	
PROJ_HDS_DEVT	Project HDS and Development Test Traceability Matrix	40 – Design\01 – Hardware
PROJ_DESIGN_FILE_999	Other design documents: diagrams, schematics, mechanical drawings, etc.	40 – Design\00 – Software 40 – Design\01 – Hardware
Verification		
PROJ_SW_FUNCT_VERIFPLAN	Software Functional Verification Plan	50 – Verification\00 – Software
PROJ_SW_FUNCT_VERIFREP	Software Functional Verification Report	50 – Verification\00 – Software
PROJ_HW_FUNCT_VERIFPLAN	Hardware Functional Verification Plan	50 – Verification\01 – Hardware
PROJ_HW_FUNCT_VERIFREP	Hardware Functional Verification Report	50 – Verification\01 – Hardware
PROJ_SW_DEVT_VERIFPLAN	Software Development Verification Plan	50 – Verification\00 – Software
PROJ_SW_DEVT_VERIFREP	Software Development Verification Report	50 – Verification\00 – Software
PROJ_HW_DEVT_VERIFPLAN	Hardware Development Verification Plan	50 – Verification\01 – Hardware

Name Document	Description	File Location
PROJ_HW_DEVT_VERIFREP	Hardware Development Verification Report	50 – Verification\01 – Hardware
Transference		
PROJ_TRANSFER_VERIFPLAN	Project Transference Verification Plan	60 – Transference
PROJ_TRANSFER_VERIFREP	Project Transference Verification Report	60 – Transference
Configuration Management		
PROJ_BASELINE	Project Baseline	80 – Configuration Management
PROJ_COMP_CNF	Project configuration composition	80 – Configuration Management
Change Control		
PROJ_CHANGE_CTRL	Project change control	90 – Change Control

Figure 5.4-6 represents all the deliverables generated iteratively and incrementally during phase 2 and consolidated during phase 3.

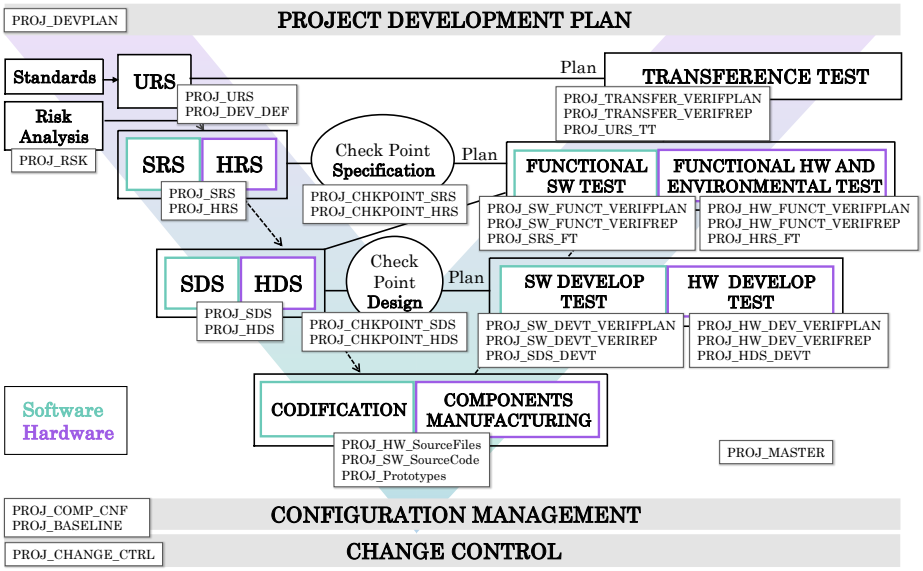


Figure 5.4-6: Deliverables generated during phase 2.

5.4.2 Project Development Plan [PROJ_DEVPLAN]

The project development process starts with project planning. The Project Development Plan includes the **information related to the management and organisation of the project during design and development** (Regul. Des. Req. 27), (Regul. Des. Req. 28).

This document should be updated as development progresses (Regul. Des. Req. 10.2). At this stage. However, the aim is not to comply with the regulatory requirements of a medical device; it is essential to have a first version of this document, as it helps the whole team to have a clear working plan.

Project Development Plan is usually drafted by the project manager, reviewed by the technical manager and approved by the client or product manager. The followings are the plans and definitions it contains as a minimum (Regul. Des. Req. 28.1):

- Responsibilities:

It is necessary to define the responsibilities of the team members. This information must be written down in the development plan and presented to the team members. In Table 5.4-2, an assignment of responsibilities is suggested.

Table 5.4-2: Principal responsibilities of each team member.

Profile	Responsibility
Technical director	- Designate a project manager. - Allocate enough appropriate technical staff (software, mechanics, electronics, testers, etc.). - Review project development plan.
Project leader	- Generate Project Development plan. - Assign software and hardware documentation manager. - Project coordination, monitoring and management. - Communication with the client or product manager.
Technicians	- Make an accurate, timely and assignable record of the design, development and verification stages.

All generated documentation must be reviewed and approved. This section should cover how the project documentation will be prepared, reviewed and approved (Regul. Des. Req. 26.2). Table 5.4-3 presents its clauses.

Table 5.4-3: Considerations for documentation review and approval.

Documentation	Responsible
Software and hardware documentation reviewer	The person in charge of software/hardware documentation will review the generated software or hardware documentation. If this person generates any software/hardware document, it will be reviewed by the project manager.
Requirements, risk analysis and project planning and management documentation reviewer	The project manager will review documentation related to user requirements, risk analysis and project management.
Document approver	The customer or product manager must approve some documents. For this purpose, they must be dated and signed by the customer or product manager. The project development plan shall indicate which documents are to be approved by the client.

- Project objectives:

The Development Plan must describe the project's objectives consistent with those described in the offer (if any).

- Coordination for the development of the project:

The project coordination conditions must be specified in detail. Table 5.4-4 shows the information to be included in this section.

Table 5.4-4: Items from the project coordination section of the Project Development Plan deliverable.

Subsection	Description
Description of participants	The activity to be carried out by each of the participants shall be described here.
Client main contacts	The contact person at the client or product manager should be defined.
Coordination procedures within the team and with the client	It should detail how the management and monitoring of the project will be carried out, and the periodicity, place, participants, and objective of these meetings should be detailed.
Planification	It includes details of (i) the phases of software and hardware development, (ii) the project schedule and (iii) who is responsible for each phase (Regul. Des. Req. 10), (Regul. Des. Req. 32).
Deliverables	Describe which deliverables will be generated due to a software or hardware development process, activity or task of the project.
Repositories	The place of storage of all the project documentation will be defined. It must be a place that ensures the traceability of the information (Regul. Des. Req. 26.3). A repository for software source code and hardware file control must be defined. It must ensure software and hardware traceability.
Reviews	Checkpoints will be defined to review the specification and design (Regul. Des. Req. 28.3). Also, the frequency of overall project progress reviews should be determined. All checks shall be recorded in minutes and stored in the project's repository.

- Equipment management:

The Project Development Plan must specify the equipment for project verifications. This equipment should be calibrated (Regul. Des. Req. 30).

- Requirement management:

This section specifies where the URS (User Requirement Specifications) will be listed. Likewise, any other document that describes the functionality or use cases of the future product must be referenced.

Furthermore, it must contain any standards, methods or tools that will be used throughout the development.

- Transfer Verification management:

This section should specify that there will be a Transfer Verification plan to be followed to check the URSs (Regul. Des. Req. 10.1). This section should detail in which document this information is stored.

In addition, a record of the execution plan shall be kept; this document shall also be mentioned here.

- Risk management:

This section should detail how the risk analysis will be performed considering the URSs and the use cases (Regul. Des. Req. 10.1). The document in which the risk analysis will be collected shall be detailed in this section.

- Specification management:

Details must be given on how the specifications are generated and collected based on the URSs and the identified risks. In this case, dividing them into HRSs and SRSs is proposed. The document that will collect this information must be detailed.

- Functional Verification management:

This section details where and how both the plan and the hardware and software functional verification record will be collected (Regul.

Des. Req. 10.1). It must be specified that these tests verify the defined specifications. The generated deliverables shall be detailed.

- Specification checkpoint management:

Details of how the correct definition of the specifications will be verified through checkpoints must be included. Also, it must be determined where the result of this review will be collected.

- Design management:

This section describes how the design is documented and performed, specifying the level of detail that will be included in the design documents.

It also specifies who is responsible for the hardware and software design components and the tools used to carry out the design (Regul. Des. Req. 10.1).

- Development Verification management:

The design should be verified with the Development Verification; this section should contain the different types of tests to be carried out and the documents in which the plan and its results will be recorded (Regul. Des. Req. 10.1).

- Design checkpoint management:

Once the design has been specified, a design review should be carried out. This section should state when such a check will be carried out and how the result will be recorded.

- Implementation management:

This section covers how the software programming and hardware construction will be performed (Regul. Des. Req. 10.1); it also details software style guides (Regul. Des. Req. 10.3).

Moreover, this section should specify how different hardware components will be built. This section should define how the quality of purchases and prototypes provided or manufactured by third parties will be guaranteed. Although they are not part of the final product, these components will be used to verify the different

functionalities, so it is essential to consider the management of suppliers (Regul. Des. Req. 29).

- Configuration management:

This section defines the documents that will contain the configuration of all the components of the project and the traceability between the different releases (Regul. Des. Req. 10.1).

- Change control:

This section should detail how changes that arise during development or after product release will be managed. This methodology considers that changes are managed through iterations. Depending on the change, the iteration will have a different scope regarding tasks and deliverables to be modified.

Changes or improvements due to errors or specification changes must be recorded in the different documents.

5.4.3 User Requirement Specification (URS) [PROJ_URS]

The User Requirement Specification (URS) **represent the user requirements and requirements derived from regulatory standards** applicable to the product to be developed (Regul. Des. Req. 1), (Regul. Des. Req. 2), (Regul. Des. Req. 5), (Regul. Des. Req. 6), (Regul. Des. Req. 9), (Regul. Des. Req. 23), (Regul. Des. Req. 28.2), (Regul. Des. Req. 31).

These URSs are based on the system's use cases and describe the device's expected performance. They **do not represent the associated technical solution**. The client or the product manager must provide the URSs at the beginning of the project as they are the critical input for the design and development of the system.

In this second phase of the methodology, not all requirements are generally precise. However, it will be **important to define the core requirements of the system**. Usually, the requirements that can be specified are those related to the device's core, the unit of measurement or monitorization. Therefore, at least these requirements should be included in the associated deliverables. Also, **obtaining the regulatory requirements during the early stages**

of the development will be necessary since, as discussed in the previous sections, these standards affect the extraction of specifications for the project's different parts.

The document [PROJ_URS] will include the user requirements. This document will be referenced in the master document [PROJ_MASTER]. This methodology proposed to define URSs in table format (see Table 5.4-5). Each URS is composed of the following:

- Identifier: It is a coding that will uniquely identify each requirement. It is defined as URS_XXX_YYY_999, where XXX and YYY are two nouns related to the requirement to know to which functional block it corresponds. The 999 will be a number used to enumerate the different requirements. Dividing the URS into blocks and sub-blocks will help the team to undertake the development in parts and thus manage the development process more efficiently (Methodol. Req. 1).
- Name: The name given to the requirement.
- Description: The description or definition of the requirement.

Table 5.4-5: URS definition.

Identifier	Name	Description
URS_XXX_YYY_999		

In addition to the URS document, it is recommended to have a **specification document** [PROJ_DEV_DEF] **detailing the use cases of the product**. As described above, the URSs will derive from these use cases.

Both [PROJ_URS] and [PROJ_DEV_DEF] must be generated by the client or product manager and reviewed and accepted by the project manager.

5.4.4 Transference Verification Plan [PROJ_TRANSFER_VERIFPLAN]

Once the URSs have been defined, the transference verification plan [PROJ_TRANSFER_VERIFPLAN] has to be determined (Methodol. Req. 3). The **transference is understood as the stage at which the**

project is to be handed over to the client or product manager (Regul. Des. Req. 28.4).

The purpose of the **transference verification plan** must fulfil several objectives. Firstly, this plan **ensures that the client or product manager accepts the development**. This stage is critical in projects where the development is carried out in external entities such as research centres or engineering companies. Although the project's acceptance criteria are usually included in the offer, this document details these criteria regarding the URSs. Therefore, this document is generated immediately after generating the URSs deliverable, before the start of the design and development, thus avoiding progress in the development without both parties having the same project vision.

In addition, this document will cover **how the transfer to the customer will be carried out and the tests that the technical department has been unable to cover** during their transfer verifications. It is not always possible for the development team to carry out all the tests as they do not always have the knowledge or the means to perform biological or patient-related tests. Therefore, this plan indicates the aspects the client or product manager should be responsible for testing during phase 3.

In addition, the **transfer tests** of the prototypes are **used to perform the formative evaluation** of the device. That is, to evaluate the usability risks involved in the device (Regul. Des. Req. 25). These tests can be performed by technical staff not involved in the development or by the client or end user.

The transference plan should be written by the development team and the project manager, reviewed by the company's technical director and accepted by the customer or product manager. The Transference Verification Plan document will be composed of three main sections:

- Development Transference Verification Plan:

Development Transference Verification Plan will cover the tests the developers can perform with the available resources. These **tests** are **used to check the correct performance of the product at the system**

level, using the latest version of the developed hardware and firmware/software.

Each transference test must have a name with the following format: TT_XXX_YYY_999, where XXX and YYY are two nouns related to the test to know to which functional block it corresponds. The 999 will be a number used to enumerate the different tests.

Transference Verification tests are defined with four fields: (i) the purpose of the test, (ii) a clear description of how to perform the test, (iii) the configuration to be used for the execution of the test, e.g., how the device should be configured, and (iv) test acceptance criteria. Table 5.4-6 presents the proposed format for defining a Development Transference Verification test.

Table 5.4-6: Transference test definition.

TT_XXX_YYY_999: Test name	
Objective	
Description	
Configuration	
Acceptance Criteria	

- Client Transference Plan:

The **Client Transference Plan** section will cover **how the project will be delivered to the client or product manager**, detailing how the deliverables will be exchanged and the deadline for the client to accept these documents. These deliverables, depending on the project of its phase, could be documents, physical prototypes, software/firmware, electronic schematics, mechanical designs, etc.

- Client Transference Verification Plan:

Client Transference Verification Plan will consist of two parts. On the one hand, the **project acceptance plan** is set out. Usually, the project manager will agree with the client on how it is accepted using what is written in the offer as a reference.

On the other hand, the Client Transference Verification Plan will be included. For this, the **developer will detail the tests not covered in their verifications**. It will be the **client's responsibility to comply**

with them. The technical department shall request the client or product manager to execute and provide a record of these tests within a certain period. Although the client must be told which tests are not covered, it will be left up to the client to define the specific tests and the deliverable format.

- URS and Transference Test Traceability Matrix:

Generating a **traceability document between** the defined **URS** and the detailed tests for the **Development Transference Verification** Plan and the **Client Transference Verification Plan** is essential.
This traceability is collected in the transference traceability matrix document [PROJ_URS_TT]. As presented in Table 5.4-7, defining the URS identifier, the URS name and the associated transfer tests is proposed. Typically, there will be several transfer tests to verify each URS.

Table 5.4-7: URS and Transference Test Traceability Matrix definition.

Identifier	URS Name	Transference Test Correlation
URS_XXX_YYY_999		TT_XXX_YYY_999

5.4.5 Risk Analysis [PROJ_RSK]

The project team will **identify and analyse potential technical failures** that can negatively impact the product's functionality (Regul. Des. Req. 8).

Risk analysis will be performed **according to ISO 14971** (Regul. Des. Req. 18), (Regul. Des. Req. 20), (Regul. Des. Req. 24), (Regul. Des. Req. 40).

This study will be collected in the document [PROJ_RSK], which will be added to the analysis performed by the client or product manager to constitute a complete risk analysis. It will occur during the project's initial phase after identifying the URSs.

The result of the risk analysis will affect SRS and HRS extraction (Regul. Des. Req. 11.4). Although **a first risk analysis is carried out at the beginning** of the development, the **risks must be re-analysed each time a change is generated** (Regul. Des. Req. 11.2).

Before the risk analysis of the design and development stage of the product can be performed, the customer or product manager must **define the intended use** of the device (Regul. Des. Req. 20.1), as well as the **safety-related characteristics** of the device, that is, known and foreseeable hazards of the product (Regul. Des. Req. 20.2)

The risk analysis **process starts** with the **identification of hazards and hazardous situations**. The risk analysis must consider the risks associated with the cybersecurity of the equipment, software development, hardware, usability, etc. Therefore, it will be necessary to **identify the hazards and hazardous situations** for each safety-related item (Regul. Des. Req. 33.1). Then, the **damage and associated causes are identified**, and the **risks are assessed**. **Risk minimisation measures** are proposed and **implemented**, and **evaluated** when required.

This process involves both the technical team and the project manager. The technical manager must review the deliverable, and the client or product manager must accept it.

The following are the main steps to perform the risk management of the product to be designed and developed:

- Identification of hazards and hazardous situations:

The project team is responsible for identifying hazards or failure modes that may affect the product (Regul. Des. Req. 20.3). This is done by brainstorming possible hazards or failure modes and their possible causes and effects. These ideas will later be used to develop the risk matrix.

- Hazards damage identification:

Once the hazards are listed, the possible damage they can cause is identified. A hazard can cause one or more damages; likewise, different hazards can cause the same damage.

- Hazards cause identification:

After describing the damage that hazards can generate, the causes that can lead to those hazards are discussed. One or more causes can create a hazard, and the same cause can generate different hazards.

- Risk assessment:

The risk assessment is carried out by assessing the identified risks' main characteristics: severity, probability of occurrence, and detectability (Regul. Des. Req. 21). For each concept, a description and a numerical value indicating criticality must be defined. Table 5.4-8 presents a proposal for a risk assessment.

Table 5.4-8: Proposed assessment of severity, occurrence and detectability.

Description	Value
Severity	
Minor	1
Moderate	2
Major	3
Catastrophic	4
Occurrence	
Remote	1
Occasional	2
Frequent	3
Detectability	
High	1
Moderate	2
Low	3
Very low	4

Once each characteristic is evaluated, the Risk Priority Number (RPN) is calculated. This value is a numerical assessment of each risk. It is calculated by multiplying the three terms, severity, detectability, and occurrence:

$$RPN = Severity \times Probability \times Detectability$$

Once the RPN for each risk has been obtained, it must be decided which risks are acceptable and which are not. A mitigation plan must be defined for risks that are not considered acceptable.

A proposed assessment of the RPNs is presented in Table 5.4-9. In green are shown the RPN values whose risk is considered acceptable (RPN less than 9). In yellow are presented those risks that the team should review to evaluate whether or not they need mitigation action (RPN greater than or equal to 9 and less than 24). Finally, in

red are those risks that will require the execution of a mitigation plan to lower their RPN (RPN greater than or equal to 24).

Table 5.4-9: Proposal for RPN assessment.

				Occurrence		
Severity		**Detectability**		Frequent	Occasional	Remote
				3	2	1
Catastrophic	4	Very low	4	48	32	16
	4	Low	3	36	24	12
	4	Moderate	2	24	16	8
	4	High	1	12	8	4
Major	3	Very low	4	36	24	12
	3	Low	3	27	18	9
	3	Moderate	2	18	12	6
	3	High	1	9	6	3
Moderate	2	Very low	4	24	16	8
	2	Low	3	18	12	6
	2	Moderate	2	12	8	4
	2	High	1	6	4	2
Minor	1	Very low	4	12	8	4
	1	Low	3	9	6	3
	1	Moderate	2	6	4	2
	1	High	1	3	2	1

- Risk mitigation and re-evaluation:

Mitigation measures are actions intended to repair or reduce the unavoidable damage generated by the future product. In this way, it is possible to specify the actions required to minimise the causes that have led to the damage.

A risk associated with a hazard may have one or more mitigation plans if so decided. In addition, a mitigation plan that applies to one

risk can be used to mitigate another risk without this mitigation plan being designed explicitly for that risk. Therefore, at this step, mitigation plans should be defined for the risks that require them (Regul. Des. Req. 22), (Regul. Des. Req. 22.1), (Regul. Des. Req. 35.2).

After setting the mitigation plan, this should be implemented (Regul. Des. Req. 22.2). Then, the risk must be reassessed, and the RPN of each risk recalculated (Regul. Des. Req. 22.3).

This process of setting mitigation actions and reassessing the risks must be iterative until the residual risk, that is, the remaining risk after the implementation of the mitigation measures, is acceptable to the client (Regul. Des. Req. 22.5), (Regul. Des. Req. 17.4). To decide if the arising risk is acceptable or not, a benefit-risk analysis should be performed using available evidence and studies to demonstrate that the benefits prevail over the residual risk (Regul. Des. Req. 22.4).

- Failure Mode and Effect Analysis (FMEA) matrix:

All the information generated in the previous paragraphs will be collected in the risk matrix (see Table 5.4-10 and Table 5.4-11, for lack of space, each risk FMEA is divided into two tables). In particular, the following information is collected:

  - *Code:* The unique identifier for the failure. It is defined as RSK_XXX_YYY_999, where XXX and YYY are two nouns related to the risk to know to which functional block it corresponds. The 999 will be a number used to enumerate the different risks.
  - *Description:* A description of the identified risk.
  - *Cause(s) of failure:* A brief description of the events and conditions that may produce the failure.
  - *Failure mode:* It is a brief description of the actual failure.
  - *Harm:* It is the potential harm generated by the failure.
  - *Probability (P):* The probability of failure before applying the corrective actions. Valid values are proposed in Table 5.4-8.

- *Severity (S):* The failure severity before applying the corrective actions. Valid values are proposed in Table 5.4-8.
- *Detectability (D):* It is the detectability of the failure before applying the corrective actions. Valid values are proposed in Table 5.4-8.
- *RPN:* The Risk Priority Number. It is calculated by performing the product of Probability x Severity x Detectability.
- *Mitigation:* It is the control measure or corrective action to reduce the probability or detectability.
- *Residual Probability (RP):* It is the probability of failure after the corrective actions are applied. Valid values are proposed in Table 5.4-8.
- *Residual Severity (RS):* The failure severity after the corrective actions are applied. Valid values are proposed in Table 5.4-8.
- *Residual Detectability (RD):* It is the detectability of the failure before applying the corrective actions. Valid values are proposed in Table 5.4-8.
- *Residual RPN (R_RPN):* It is the risk priority number of the residual risk.

Table 5.4-10: Risk matrix (before mitigation actions).

Code	Description	Cause	Failure Mode	Harm	P	S	D	RPN
RSK_XXX_ YYY_999								

Table 5.4-11: Risk matrix (after mitigation actions).

Code	Description	Mitigation	RP	RS	RD	R_RPN
RSK_XXX_ YYY_999						

5.4.6 Software [PROJ_SRS] and Hardware [PROJ_HRS] Requirement Specification

Based on the risk analysis and the URSs, Software Requirement Specification (SRS) (Regul. Des. Req. 11), (Regul. Des. Req. 33) and

Hardware Requirement Specification (HRS) must be extracted. The **SRS indicate what software must do** to meet the user requirements or risk analysis. On the other hand, the **HRS mean what needs to be done by hardware** to meet the requirements. Therefore, these requirements will not only **contain functionality requirements** but **also define requirements related to the safety and security** of the device (Regul. Des. Req. 33.1).

The first step is to **extract the high-level specifications** to define the requirements better. Based on the document [PROJ_DEV_DEF] that defines the product's use cases, the high-level functional and non-functional requirements for both software and hardware are identified. Once this first definition is done, the extracted requirements are refined and **broken down into a list of detailed requirements**. **The refinement of the SRS and HRS will increase as the project progresses through the different iterations and phases**.

SRSs must be generated by the software team, and HRSs by the hardware team. Both are reviewed by the project manager and approved by the client.

This methodology proposes its definition in a table format. Table 5.4-12 contains an example of the proposed format. Although Table 5.4-12 presents HRS and SRS in the same table, generating a separate document for each is suggested. Its main fields are identifier, name, description and related URS or risk.

- Identifier: It is a coding that will uniquely identify each requirement. It is defined as SRS_XXX_YYY_999 or HRS_XXX_YYY_999, where XXX and YYY are two nouns related to the specification to know which functional block it corresponds. The 999 will be a number used to enumerate the different specifications.
- Name: It is the name given to the specification.
- Description: It is the description or definition of the specification.
- URS or Risk correlation: The URS or risk from which the specification is derived.

Table 5.4-12: SRS and HRS definition.

Identifier	Name	Description	URS or Risk correlation
SRS_XXX_YYY_999			RSK_XXX_YYY_999
HRS_XXX_YYY_999			URS_XXX_YYY_999

5.4.7 Software [PROJ_SW_FUNCT_VERIFPLAN] and Hardware [PROJ_HW_FUNCT_VERIFPLAN] Functional Verification Plan

The **functionality of a system is verified with Functional Testing**. Although run against the whole system, each test can only verify a part of the functionality (Regul. Des. Req. 16).

The tests and their acceptance criteria **will be defined according to the SRSs and HRSs** (Regul. Des. Req. 16.1). These tests are defined just after the definition of the requirements, before the implementation of the design stage (Methodol. Req. 3).

Usually, Functional Testing is **focused on verifying "what" the system does** and not "how" it is done. Nevertheless, Functional Testing should follow a black-box approach. In addition, this stage should **also include tests** related to verifying the **environmental performance** of the designed hardware and **cyber-security** tests (Regul. Des. Req. 38.1).

The Functional Verification Plan must list the individual functional tests that will be used to check portions of the whole system's functionality. **Each functionality may be verified through one or multiple tests**. The description of the tests must include the functionality to be tested, system configuration, inputs and acceptance criteria. Specifically, each test must consist of the following information:

- Identifier: A unique test identifier formatted as FT_XXX_YYY_999 where XXX and YYY are two nouns related to the test to know to which functional block it corresponds. The 999 will be a number used to enumerate the different tests.
- Test name: It is the name of the test.

- Objective: The purpose of this specific test.
- Description: A description of the test to perform. Ideally, it will include "what" will be tested (e.g., function, functionality, range, etc.) and the process that will be followed.
- Configuration: It includes all the relevant equipment, tools, third-party software, sets of software configuration options and, in summary, any relevant contextual configuration that impacts the test or its performance.
- Acceptance Criteria: A set of conditions a component or system must satisfy to be accepted as "completed" or "successful". Acceptance criteria should focus on the "what" and avoid the "how".

Table 5.4-13 presents the proposed software and hardware functional test format.

Table 5.4-13: Functional test definition.

FT_XXX_YYY_999: Test name	
Objective	
Description	
Configuration	
Acceptance Criteria	

All SRS and HRS must be verified with one or more functional tests. The **relationship between SRS/HRS and functional tests** is **listed in a traceability matrix**. SRS and functional software tests traceability is collected in [PROJ_SRS_FT], and for HRS and hardware functional tests in [PROJ_HRS_FT], a similar table format is proposed for both cases. Table 5.4-14 presents the fields that constitute this matrix, the SRS/HRS identifier, the name of the SRS/HRS and the associated functional test.

It is recommended that the **tests be executed by testers who are not involved in developing the medical device's hardware or software**; this helps to **avoid any conflict of interest**.

Table 5.4-14: SRS/HRS and Functional Test Traceability Matrix definition.

Identifier	SRS/HRS Name	Functional Test Correlation
SRS_XXX_YYY_999		FT_XXX_YYY_999
HRS_XXX_YYY_999		FT_XXX_YYY_999

5.4.7.1 Environmental safety test verification

The hardware of the final product must be tested according to functional safety regulations to achieve CE marking. Although this validation is contemplated during phase 3, it is **good practice to carry out environmental pre-certification tests** during phase 2 to **characterise the design and minimise the probability of failure** once it is **in official laboratories**. Therefore, at this stage, together with the functional verification tests, it is proposed to carry out environmental safety pre-certification tests (Regul. Des. Req. 7).

This family of tests includes electromagnetic compatibility, electrostatic discharge, temperature, vibration, etc., tests explicitly identified in the HRSs. This verification stage is closely linked to specific test equipment. Its use must be foreseen and described in the verification plan.

5.4.7.2 Cybersecurity test verification

Cybersecurity Testing is a particular Functional Testing case in which security risks are specifically targeted. Cybersecurity testing is often driven by a normative (such as UL2900-2-1) or good practice guides.

Nevertheless, cybersecurity tests **should consider all the system's external interfaces and domains**, such as data integrity, privacy, accessibility and authentication. Additionally, cybersecurity tests **should cover software coding weaknesses** introduced by the development process or potentially harmful external actors.

5.4.8 Software [PROJ_CHKPOINT_SRS] and Hardware [PROJ_CHKPOINT_HRS] Specification Checkpoint

Once the functional tests for the defined SRS and HRS have been determined, the proposed methodology contemplates some **control**

points to check that these requirements are complete, verifiable, consistent and traceable (Methodol. Req. 28).

Specifically, the following aspects should be reviewed at this control milestone:

- All defined URS and mitigation actions described in the risk analysis are translated into software or hardware requirements. That is, they have an associated SRS or HRS.
- The list of SRS and HRS is enough to cover the risk analysis's URSs and mitigation actions.
- The defined SRS and HRS are consistent, and there is no contradiction.
- Each SRSs and HRSs has at least one functional test that verifies it.
- The technical requirements associated with embedded systems have been addressed (Techn. Des. Req. 1 - 11).

The project manager will be responsible for this control milestone. The result, conclusions and defined actions will be written down in the corresponding software [PROJ_CHKPOINT_SRS] or hardware [PROJ_CHKPOINT_HRS] report. In addition, the **review of safety and security requirements will be carried out by experts in the field** (Regul. Des. Req. 33.2).

5.4.9 Software [PROJ_SDS] and Hardware [PROJ_HDS] Design Specification

The SRS and HRS must be translated into design specifications for software, **Software Design Specification (SDS), and hardware, Hardware Design Specification (HDS).** These design specifications **specify how to solve each SRS and HRS**. These specifications **address high-level and detailed design** (Regul. Des. Req. 13).

Regarding **software**, this stage includes the design of a **high-level architecture** to define the deployment of the system and **its breakdown into components** (Regul. Des. Req. 12), (Regul. Des. Req. 34). This architectural design must be carried out from a functional and safety/security point of view (Regul. Des. Req. 34.1). Likewise, the interfaces and protocols used to communicate between them will be defined.

Detailed design will provide **structure and detail to each high-level component** (Regul. Des. Req. 13.1), (Regul. Des. Req. 13.2). Additionally, it must have enough segregation to control potential risks (Regul. Des. Req. 12.3), (Regul. Des. Req. 35), (Regul. Des. Req. 35.1).

For both types of design, graphical design tools can be used (Methodol. Req. 24), providing different system perspectives through diverse views and diagrams. Each perspective approaches the system from another point of view. For instance, possible diagrams representing software could be system deployment diagrams, sequence diagrams, state machine diagrams, etc. (Regul. Des. Req. 12.1).

Regarding **hardware**, **a representation of the envisaged technological solution must be provided**; this shall include the main processes, signal flows, internal and external interfaces, and system key components. The product's life cycle, including verification and industrialisation, shall be considered to design the required components.

The **high-level design must contemplate all hardware components** (electronic and mechanical); this **will derive into the detailed mechanical design, schematic design, and PCB** (Printed Circuit Board) design.

SDSs must be generated by the software team, while HDSs are to be developed by the hardware development team. Both must be reviewed by the project manager and approved by the client.

This methodology proposes to define the SDS and HDS in a table format, including the following fields:

- Identifier: An unambiguous coding for each design specification defined as SDS_XXX_YYY_999 or HDS_XXX_YYY_999 where XXX and YYY are two nouns related to the functional block that the design belongs. The 999 will be a number used to enumerate the different design specifications.
- Name: A name given to the design.
- Description: The description or definition of the design.

- Context: The documents or diagrams that serve as a context for defining design specifications, such as high-level diagrams, schematic diagrams, flow diagrams, etc.
- SRS/HRS correlation: The SRSs or HRSs from which the design is derived.

Table 5.4-15: SDS and HDS definition.

Identifier	Name	Description	Context	SRS/HRS correlation
SDS_XXX_ YYY_999			Document: www Chapter: xxx File: yyy Diagram: zzzz	SRS_XXX_ YYY_999
HDS_XXX_ YYY_999			Document: xxx Chapter: yyy	HRS_XXX_ YYY_999

The SDS table shall be contained in document [PROJ_SDS] and the HDS table in document [PROJ_HDS]. All SRSs and HRSs must be represented in the corresponding SDS and HDS. Furthermore, an SRS and HRS can have more than one SDS or HDS associated with them.

5.4.10 Software [PROJ_SW_DEVT_VERIFPLAN] and Hardware [PROJ_HW_DEVT_VERIFPLAN] Development Verification Plan

Once the design of the components is completed, the **Hardware and Software Development Verification Plan** is defined (Methodol. Req. 3). This plan aims to **verify the performance of each designed component** (Regul. Des. Req. 28.4). It has five main sub-plans:

- Software Unit Test Verification Plan: Each component's verification will be defined using unit tests (Regul. Des. Req. 14.2).
- Software Integration Test Verification Plan: This plan will define how to integrate the components and how to verify that their integration interfaces comply with the design specifications (Regul. Des. Req. 10.3), (Regul. Des. Req. 15), (Regul. Des. Req. 15.1), (Regul. Des. Req. 37).
- Software Static Verification Plan: This plan will define how to perform the static verification of the source code.

- Hardware Unit Test Verification Plan: This plan envisages the unitary verification of the different blocks that build up the hardware of the system
- Hardware Integration Test Verification Plan: It defines the verification of the electromechanical integration of the developed device.

For both software and hardware, the corresponding development verification plan document [PROJ_SW_DEVT_VERIFPLAN] and [PROJ_HW_DEVT_VERIFPLAN] must be generated. This document shall have differentiated sections for each of the test types.

This document shall be generated by the development team, reviewed by the project manager and approved by the client. In general, tests will be defined in a table format by filling in the following fields:

- Identifier: A unique test identifier formatted as DevT_XXX_YYY_999 where XXX and YYY are two nouns related to the test to know to which development block it corresponds. The 999 will be a number used to enumerate the different tests.
- Test name: The name of the test.
- Objective: The purpose of this specific test.
- Description: A description of the test to perform.
- Components under Test: It specifies the software or hardware component to be verified.
- Test Configuration: It contains relevant aspects to reproduce and carry out the tests. e.g. test conditions, test procedures, etc.
- Acceptance Criteria: The set of conditions that a component or system must satisfy to be accepted as "completed" or "successful".

Table 5.4-16: Development test plan definition.

DevT_XXX_YYY_999: Test name	
Objective	
Description	
Components Under Test	
Test Configuration	
Acceptance Criteria	

The development tests **shall verify each of the defined SDS and HDS**. One verification test can verify more than one SDS or HDS, or several tests can verify a single SDS or HDS. The **relationship between the SDS/HDS and the tests that verify them must be documented**. This is done through the software [PROJ_SDS_DEVT] and hardware [PROJ_HDS_DEVT] design and verification traceability matrixes.

Table 5.4-17 presents the format of these traceability matrixes.

Table 5.4-17: SDS/HDS and Development Test Traceability Matrix definition.

Identifier	SDS/HDS Name	Development Test Correlation
SDS_XXX_YYY_999		DevT_XXX_YYY_999
HDS_XXX_YYY_999		DevT_XXX_YYY_999

5.4.10.1 Software Static Verification

Static Testing groups **a series of software testing procedures that do not involve the execution of the source code**.

The main method used in Static Testing is the **Peer Code Review**. This method **inspects and matches the software code and documentation against the system requirements**. The code inspection includes tests associated with the code that are code in themselves (e.g., unit or integration tests).

Each code entry must be matched to a system requirement and vice versa; this ensures that all the requirements are implemented in the

code and that there is no potentially dangerous superfluous, or useless code.

One of the main characteristics of the Peer Code Review method is that **the person(s) reviewing the code MUST not be the same one(s) that created the code**.

Optionally, **Static Analysis tools could be used to analyse the code base**. Static Analysis tools **enable automatic source code inspection** and readily find common errors like buffer overflows or uninitialized variables and more complex ones like variable race conditions.

5.4.10.2 Software Unit Test Verification

Unit Testing refers to the **tests performed on the SW components that comprise the system**. The tests to be committed to each element and the acceptance criteria are derived from the SW component specification.

Although various methods could be used to perform Unit Testing, the preferred method is Unit Tests and Unit Test Harnesses (Methodol. Req. 24). Table 5.4-18 presents some Unit Test Harnesses depending on the target platform and programming language.

Table 5.4-18: Unit Testing Harnesses.

Harness	Platform	Language
CppUTest [319]	Linux embedded	C, C++
C Unity [320]	Non-Linux embedded	C

It is also possible, and sometimes even recommended, to use dual-platform and cross-platform Unit Test approaches. These approaches are based on using one platform type to implement the Unit Tests, different from the platform used to run the Unit Tests.

The use of simulators for preliminary debugging and verification is also encouraged. This approach has the advantage that a more powerful and resourceful platform can be used to develop the tests and preliminarily verify the code before performing the tests in the target platform. However, note that the **relevant Unit Tests are performed in the target platform**.

5.4.10.3 Software Integration Test Verification

Integration testing is used to verify that independently developed components are grouped correctly and behave according to the specifications. Parts are usually integrated iteratively and incrementally. Integration testing should be planned accordingly, considering the architecture defined in the SDS. Additionally, **each Integration step/iteration must be verified against the acceptance criteria derived from the SDS**.

An integration plan should be produced and periodically revised to manage the integration process. The integration plan should consider several aspects, such as development priorities, timelines, uncertainties, availability of resources, etc.

Unit Test Harnesses can be used to automate integration tests. Furthermore, mock components can be used to satisfy the dependencies and conditions of the elements that are currently being integrated.

5.4.10.4 Hardware Unit Test Verification Checkpoint

This phase concentrates on **testing the different electronic and mechanical blocks that make up the hardware design of the solution**. Typical blocks to be tested at this level include power supplies, communication buses, analogue inputs and their filtering and digitising stage, digital inputs, clocks, memories, available peripherals, etc.

In this case, **the order of execution of the tests is very relevant** since the **unsatisfactory execution of any tests could lead to irreparable damage** to the elements to be tested or associated. Likewise, recommendations for prior checks to be carried out are included.

The verification plan for this level should include the list of elements to be tested and should be as complete as possible to try to avoid unperformed checks at a later stage.

It is important to emphasise that this **hardware block verification stage depends on an FW/SW that allows activating and deactivating parts of the hardware** and its peripherals. The verification plan shall

describe the scope of such a FW/SW. The standard order to be followed for the verification shall be as follows:

- Preliminary checks:

These verifications are carried out before the voltage is supplied to the circuit. It is necessary to conduct verifications such as a visual inspection of the PCB, verifying the correct assembly of optional elements, measuring voltage levels, adjusting resistors' value and proving that there is no short-circuit in any power supply voltage.

- Power supplies and power consumption:

All the system's power supplies must be checked within the operating limits defined by their respective datasheets. It is also necessary to check the power consumption of each power source.

- System clock:

Verify that the system clocks are within the defined nominal limits for frequency, waveform, and voltage levels.

- CPU basic activations:

CPU testing has different levels depending on the design and complexity. However, it is recommended to verify at least the programming port detection and the downloading and execution of the FW/SW.

- Peripherals:

The verification focuses on checking each of the peripherals included in the system individually. The checks are based on discerning whether or not the CPU can drive these peripherals and whether or not the signals reach the corresponding terminals correctly.

- Communication buses:

In this case, it is necessary to check that communication buses operate within the defined level and speed limits, especially for asynchronous buses such as UART. For higher-speed buses or buses requiring precise test instrumentation (e.g. RAM buses, MIPI buses,

Ethernet buses, USB buses), a specific test strategy, which is often functional-level, must be considered.

- Analogue circuitry:

Filters, ADCs, and DACs will be tested according to the design they respond to, typically defined in detail in the HDS. In this case, it is relevant to have the appropriate equipment for signal generation, to inject it into the input channels and to be able to perform meaningful and deterministic tests of the acquisition channels.

- Memories:

Designs typically include non-volatile (Flash, EEPROM) and volatile (RAM, DDR, DDR2, LPDDR) type memories. Although the memory acts as a data repository, it is also a storage element for executing the software program. It is common for the software to be stored in a Flash-type memory. When the computer starts up, the CPU automatically reads its contents and dumps them into the RAM for execution. Because of this, basic memory verifications are assumed within the standard CPU start-up tests.

- Battery subsystem:

Different types of verifications are considered for this subsystem: verification of the voltage and current profiles on charge and discharge, verification of the storage capacity for the expected rate of use, verification of the protection systems, verification of the system behaviour under different temperature conditions; and verification of the accuracy of the state of charge estimation system (gauge).

- Inspection of mechanical prototypes:

The person responsible for the mechanical design must carry out an isolated verification of the mechanical part by visually inspecting the manufactured mechanical parts.

Specifically, it will be necessary to measure the different parts of the manufactured prototype, checking that they are within the defined tolerances. It will also be required to check the quality of the mechanical parts, materials, painting processes, etc. Any non-

conformity will have to be recorded during the execution of the verification.

5.4.10.5 Hardware Integration Test Verification

This verification phase **checks that the electronic parts** (PCBs, pushbuttons, LEDs, connectors, etc.) **fit correctly into the designed and manufactured mechanical parts**. It shall be checked that the environmental protection solutions, fastening solutions, connectronic solutions, button solutions, pushbuttons, LEDs and the handling and assembly operations of the prototype are under the defined HDS.

5.4.11 Software [PROJ_CHKPOINT_SDS] and Hardware [PROJ_CHKPOINT_HDS] Design Checkpoint

Once the high-level design, the detailed design and the development verification plan have been defined, the **design** checkpoint is executed to review the architectural and detailed design (Methodol. Req. 28), (Regul. Des. Req. 34.2), (Regul. Des. Req. 35.3).

This checkpoint will **verify that each SRS and HRS has one or more SDS and HDS detailing the design of each requirement**. In addition, this **review will assess the technical adequacy of the design** and, if applicable, the compatibility with other hardware or software with which the future product must interface. Lastly, it must be **verified that each SDS and HDS has at least one development test** that verifies it.

Regarding software, at least the following checks will be carried out:

- The SDSs are detailed enough to be comprehensible to the developer who has to codify the software.
- The architecture and detailed design address the SDSs to provide a solution to each SRSs. It must be verified that all functional blocks are included in the SDSs or architecture diagrams and that the SDSs are supported by additional diagrams that clarify the software design.

For the hardware, the design controls are divided into four categories:

- Architecture design control: It is checked that all electronic and mechanical components and their interfaces and environmental requirements are identified and included in the HDSs.
- Schematic control: It is checked that the schematics meet the design described by the HDS. Additionally, the schematics' quality is checked to minimise the probability of errors in the following stages. Verifications of libraries, CPU's pin functions, design of all functional blocks, etc., are considered.
- Mechanical design control: It controls that the mechanical design and the generated documentation are correct to launch the manufacturing of the components. Likewise, the documentation must be checked to be accurate to the HDSs. Some aspects, such as interferences between parts, tolerances, materials, and properties, must be considered.
- PCB design control: The layout and the generated documentation for manufacturing and assembling the designed electronics are checked. The synchronism between the schematic and PCB, verification of libraries, documentation, gaps between tracks, thicknesses, holes, etc., must be verified.

In addition, it will be necessary for both software and hardware to check that the technical design requirements associated with embedded software have been addressed (Techn. Des. Req. 1 - 11).

A positive outcome is required during this review to proceed with the implementation phase.

5.4.12 Software [PROJ_SW_SOURCE] and Hardware [PROJ_HW_SOURCE] Implementation

This stage is divided into the software implementation and the construction of the hardware components.

Regarding **software coding, product application, unit tests, and integration tests must be created** (Regul. Des. Req. 14), (Regul. Des. Req. 14.1), (Regul. Des. Req. 36). Likewise, ensuring that the whole team uses the same development tools is essential. Furthermore, good **coding practices must be followed**, including secure coding standards, style guides, use of patterns and best practices, and

modular development of the source code (Regul. Des. Req. 36.1). Finally, software source code should be stored in reposi**tories that allow good management of software traceability** and workflows.

The **construction of hardware** components is **based on the documentation generated during the design stage**. Electronics and mechanics are manufactured based on the layout and the generated documentation (BOM, gerbers, drawings, etc.).

5.4.13 Software [PROJ_SW_DEV_VERIFREP] and Hardware [PROJ_HW_DEV_VERIFREP] Development Verification Report

Once the software implementation and hardware production have been completed, the verification stage must be completed. In this stage, the **tests** defined in section 5.4.10 **are executed and recorded** (Regul. Des. Req. 15.2). The following tests are envisaged: **Software Unit Tests, Software Integration Tests, Software Integration Tests, Software Static Tests, Hardware Unit tests and Hardware Unit Tests**. All these tests will verify the software and hardware architecture, detailed design and implementation (Regul. Des. Req. 12.2), (Regul. Des. Req. 13.3), (Regul. Des. Req. 36.2), (Regul. Des. Req. 37).

This methodology proposes to record the test result in a table format (see Table 5.4-19). This table contains the following information:

- Identifier: The unique code assigned to the test must match the value set in the development verification plan.
- Test name: A descriptive name of the test, the same as defined in the plan.
- Objective: It is the purpose of the test. Equal to the one specified in the development verification plan.
- Date: It is the date when the test was performed.
- Responsible: It is the person or persons that were involved in the test and its evaluation.
- Code, hardware tag: The software or hardware versioning tag associated with the block must be verified. In the case of software, the version number must be included, and the repository in which it is stored must be referenced.

- Result: The result of the test evaluation is recommended to be OK/NOT OK, but it is also possible to collect some clarifying comments.
- Evidence: Any evidence that may prove that the test was correctly evaluated.

Table 5.4-19: Development test report definition.

DevT_XXX_YYY_999: Test name	
Objective	
Date	
Responsible	
Code, hardware tag	
Result	
Evidence	

5.4.14 Software [PROJ_SW_FUNCT_VERIFREP] and Hardware [PROJ_HW_FUNCT_VERIFREP] Functional Verification Report

After the development verification, **functional verification** must be carried out. In these tests, **cybersecurity and safety pre-certification tests** are also contemplated. At this stage, the tests defined in section 5.4.7 are recorded (Regul. Des. Req. 16.3). These tests will verify the defined SRS and HRS (Regul. Des. Req. 11.3), (Regul. Des. Req. 28.4).

As in the case of developmental tests, it is proposed to collect the test record in table format (see Table 5.4-20). The fields included are test identifier, test name, objective, date, responsible, code or hardware tag, result and evidence. The execution of functional verification **tests should be performed by people who have not contributed to the development of the test**.

Table 5.4-20: Functional test report definition.

FT_XXX_YYY_999: Test name	
Objective	
Date	
Responsible	
Code, hardware tag	
Result	
Evidence	

5.4.15 Transference Verification Report [PROJ_TRANSFER_VERIFREP]

The verification ends with the execution of the **transference tests** that **verify the defined URS and the transference of the development to the product manager or client**; specifically, the tests performed at this stage are defined in section 5.4.4.

As with the development and functional tests, the record should include the test identifier, test name, test objective, date and responsible, versioning tag for software and hardware source code, result and evidence. The format is presented in Table 5.4-21.

Table 5.4-21: Transference test report definition

TT_XXX_YYY_999: Test name	
Objective	
Date	
Responsible	
Code, hardware tag	
Result	
Evidence	

5.4.16 Configuration Management [PROJ_COMP_CNF] and Baseline [PROJ_BASELINE]

In this stage, all the **critical building blocks of the development are identified**; this includes defining the product structure, breaking it down into configuration elements, describing the composition of the configuration of each part and defining the baseline.

First, the project development team must define the p**roduct's structure by dividing it into the elements** necessary to achieve the specified functionality. **Each defined element's configuration composition shall be filled in**, indicating the characteristics for generating that element.

As for the software developed, at least the following elements should be considered (Regul. Des. Req. 19), (Regul. Des. Req. 41):

- Development tools (development environment and compiler versions).
- Development and execution of the operating system.

- Programming language.
- Source code.
- Used third-party drivers or libraries.
- The input files required for each element.

In terms of hardware, at least the following elements should be covered:

- Development tools: The software such as the one used for generating schematic, PCB, mechanics, optics and other hardware elements. The version of each of them must be collected.
- Hardware elements that make up the development: It is necessary to include their version and identification.

In addition, the **baseline must be defined**, **showing the state of each element** that **makes up the system at a given time**. The baseline must contemplate software, hardware, documents, etc. All these elements must be versioned. **When a project delivery is generated, it will be necessary to define a baseline that identifies the status of all elements at that moment**.

In this methodology, it is proposed to record this information in a table format (see Table 5.4-22). Each table entry represents a specific instant of the elements that make up the system, and all the lines represent the evolution of the product over time. It is proposed to include at least the following information:

- Id baseline:

An identifier is assigned to the baseline, such as BL_YYYYMMDD, where BL refers to the baseline entry, YY is the year it is generated, and MM is the month and DD the day.

- Date:

The date the table entry or project release is generated.

- Comments:

Additional information about the delivery, e.g. "final delivery of the project".

- Hardware elements:

A table entry must be generated for each hardware element (mechanics, electronics, optics, etc.). In this line, it will be necessary to unequivocally identify the component's version at the moment of the release.

- Software elements:

As for hardware components, for each software element (program), an entry must be generated in the table. This element can be identified by software version and git commit.

- Requirements:

It includes documents containing user requirements. The URS document and its version should at least be referenced here.

- Specifications:

Software (SRS) and hardware (HRS) specification documents identified with their versions must be detailed. When more specification documents exist, they must also be collected.

- Design:

Hardware (HDS) and software (SDS) design documents must be listed. The version of these documents also needs to be detailed.

- Verifications:

Verification plans and records must be identified at all levels, transfer, functional and development, including their version.

- Risk management:

The design and development risk analysis specifying its version must be included.

- Others:

Other documents or resources generated in the project, such as the configuration composition document, can be included in this section.

Table 5.4-22: Baseline definition.

	ID baseline	BL_YYMMDD
	Date	
	Comments	
Hardware (element A)	Version	
Hardware (element B)	Version	
Software (element A)	Version	
Software (element B)	Version	
Requirements	URS documents	
Specifications	SRS/HRS documents	
Design	SDS/HDS documents	
Verification	Transference, functional and development test documents	
Risk management	RSK document	
Others	Additional elements and documents	

The hardware and software development team will generate the configuration composition. As for the baseline, the project manager will generate a new entry at release time.

5.4.17 Change Control [PROJ_CHANGE_CTRL]

There could be **change requests for both software and hardware during development and after product release**. These requests could be due to modifications for improvements, updates or problem-solving, among others.

The **client or product manager must request changes to fix errors, make improvements, and perform updates**. The development team will not be responsible for deciding what changes to push and when to make them.

Clients, technical assistance services or product managers usually report problems identified in the field. Once received, the client must evaluate them to determine whether they are considered errors. If it is an error, a report of the failure must be generated. This information must help the developer to replicate the error.

Once the error has been registered or the change or improvement identified, the development team must generate a change request;

the project manager must approve this request. Based on the record provided by the client, the project manager will assign the change to the appropriate development team.

In case **the change implies a new requirement or design change, this will be reflected in the respective documents**: URS, SRS/HRS (Regul. Des. Req. 11.1), SDS/HDS, Risks (Regul. Des. Req. 11.2), verifications (Regul. Des. Req. 16.3), etc. **Depending on the performed change, it is necessary to go through one or more development process phases** (see Figure 5.4-7).

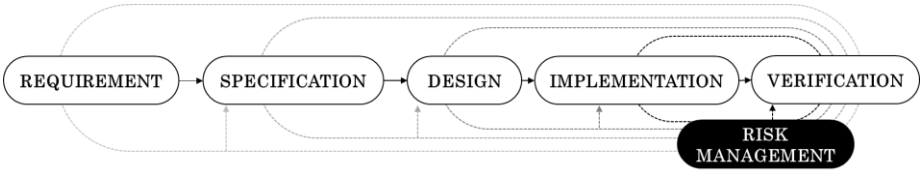


Figure 5.4-7: Diagram of processes to be re-executed after design changes.

If the change is due to an error, the record will be added to the error list. In addition, any changes made in product development must be assessed according to the risk management detailed in section 5.4.5 (Methodol. Req. 8).

5.4.18 Incremental and Iterative Prototyping Checkpoint

Phase 2 ends when the V-model is completely iterated with all the activities of the product backlog. After this execution, the checkpoint or gate of phase 2 must be executed (Methodol. Req. 28).

In this checkpoint, the **objective achievement is evaluated, and a decision is taken on whether to continue the development**. If it is decided to continue, it should move on to phase 3 to consolidate the medical product.

This decision is usually taken by the client or product manager together with the company executive team. When deciding, the company's upper management usually considers technical, economic and market aspects.

5.5 Phase 3 – Medical Product Consolidation

The third stage of the methodology aims to **consolidate the development and convert it into a medical product**. To this end, the deliverables and design generated during phase 2 are used as input. In this stage, an **intensive verification is carried out** to **consolidate and validate** both the **development and the generated documentation**. The key characteristics, inputs and outputs of phase 3 are illustrated in Figure 5.5-1.

Figure 5.5-1: Medical Product Consolidation phase: key characteristics, inputs and outputs.

Although a design with advanced functionality is available as input to this phase, it **is expected that the client or product manager wants to introduce changes** or improvements that have been identified after the delivery of phase 2. Likewise, to validate the development, **the design must consider aspects related to its industrialisation**. Therefore, in this phase, the following tasks will be undertaken:

- Minor modifications or improvements:

Although the main development is completed, minor changes or improvements must be accepted during this phase, such as changes in the colours of the graphical user interface, solutions to minimise risks identified after phase 2 development, etc.

- Industrialisation-related functionality:

During this phase, requirements related to the industrialisation of the product will be added. For this purpose, it will be necessary to involve the company's production department or, in the case of the external industrialisation of the product, the Electronic

Manufacturing Services (EMS) company responsible for this activity.

The industrialisation requirements will be considered an additional use case to the existing ones. Therefore, its requirements will be added to the URS of the product and will be considered in the SRS/HRS, SDS/HDS, implementation and verification stages. In addition, the project development plan must be updated to indicate how this stage will be undertaken.

- Intensive verification and validation of the device:

In this phase, an exhaustive verification of the product will be carried out. This verification must cover 100% of the functionality developed by the development team. Also, during this phase, the validation of the device is considered.

- Product release:

The project plan should contemplate how the release of the product and updates will be performed once the device is on the market.

5.5.1 Overall overview of the proposed model and its deliverables

In this phase, it is proposed to follow a V-model very similar to the one proposed in phase 2. In contrast to phase 2, an Agile, iterative and incremental model is not considered. Instead, a **sequential execution of the V-model is proposed**. This approach will ease the product certification by regulatory entities since the reference standards suggest the development of medical devices following the traditional V-model methodology (Methodol. Req. 2).

Figure 5.5-2 presents the V-model proposed for this phase. Figure 5.5-2 shows that the only difference at the model level is that the **transfer stage defined in phase 2 becomes a validation stage**.

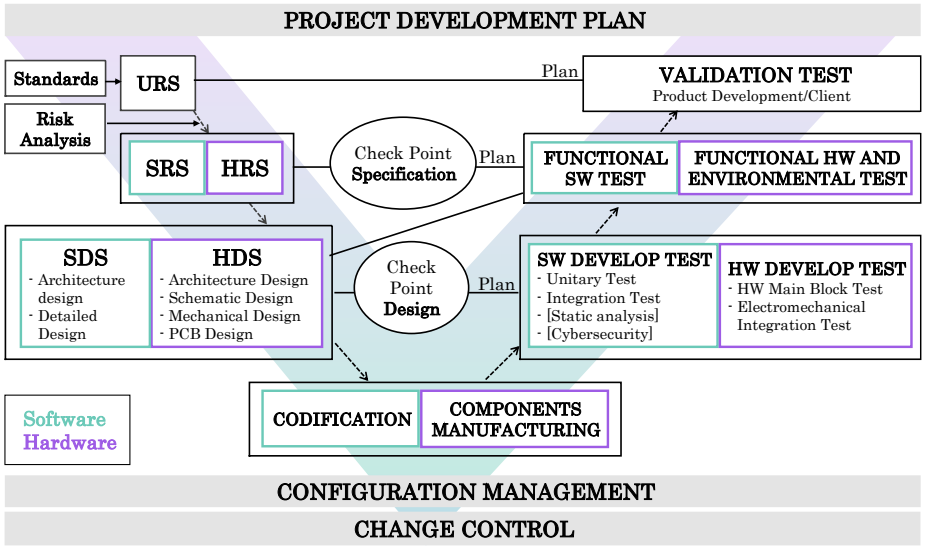


Figure 5.5-2: proposed V-model for phase 3.

The process starts with updating the project development plan generated in phase 2. This plan should be updated to include how the product industrialisation, product validation and product release will be addressed. Likewise, the roles and scope of processes should be updated where necessary. Section 5.5.2 provides more details on the changes to this deliverable.

Then, the URSs must be updated with the use cases to be implemented to undertake the industrialisation phase and the changes requested by the client or product manager.

Once the URSs have been updated, the validation plan should be generated; this plan must cover the validation of all the defined URSs. Tests related to clinical evaluation and research, or performance evaluation and performance studies, are considered at this stage. Likewise, any tests that apply to the device according to the IEC 60601 family of standards or others must be carried out in accredited entities. In section 5.5.3, the most relevant points of the validation are specified.

The risks of the product are then re-analysed, and, if necessary, appropriate control measures are taken.

Once the risk analysis has been performed, the specification stage, both for software and hardware, is carried out. This stage must

address the new functionality and use cases related to industrialisation. In addition, if necessary, the specification detailed in phase 2 can be extended to fully cover the device's specification.

After the specifications are precise, the functional and environmental verification plan for hardware and software is defined. Based on this, the specification checkpoint must be executed.

Subsequently, the high-level design and detailed hardware (HDS) and software (SDS) design are performed. The development verification plan must also be defined. Then, the development progress is verified by the design checkpoint.

The design is followed by implementing the added software and hardware functionality.

After the implementation, the defined verification plans must be executed, first the development plan, then the functional plan, and finally, the validation plan.

Finally, as in phase 2, configuration management and change control processes must be carried out.

Regarding phase 3 deliverables, Figure 5.5-3 shows that the associated **deliverables remain the same except for the validation stage**. However, **all documents must be updated according to the new functionality**. These include the deliverable project development plan [PROJ_DEVPLAN].

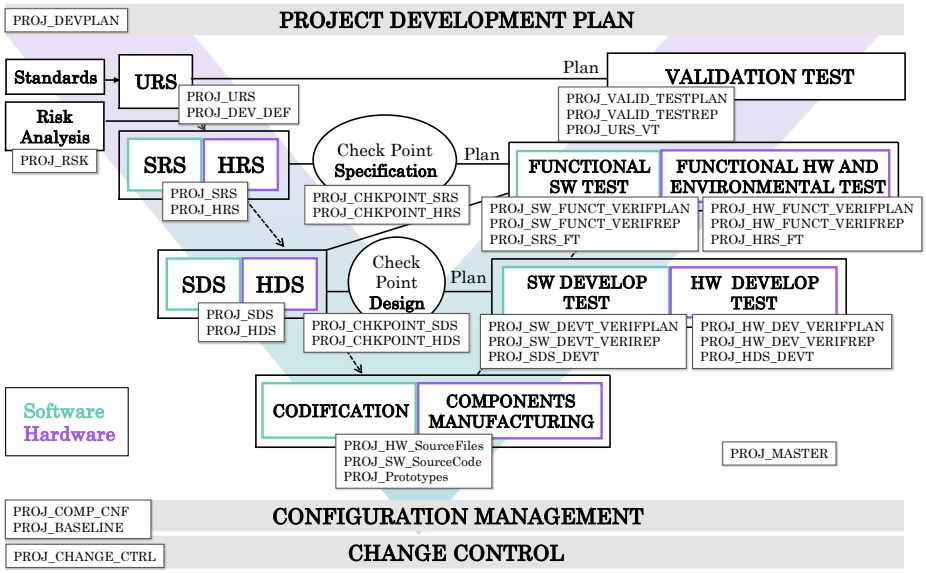


Figure 5.5-3: Deliverables associated with phase 3.

Table 5.5-1 presents the updated list and location of the final deliverables generated by the methodology proposed in this thesis. These deliverables will be listed in the master document, [PROJ_MASTER].

Table 5.5-1: Master document proposal, including all project deliverables.

Name Document	Description	File Location
Management		
PROJ_OFFER	Project Offer	00 – Management\00 – Offer
PROJ_MINUTES_YYMMDD	Project Meeting Minutes	00 – Management\01 – Meeting Minutes
PROJ_DEVPLAN	Project Development Plan	00 – Management\02 – Development Plan
PROJ_CHKPOINT_SRS	Project SRS Check Point	00 – Management\03 – Check Points
PROJ_CHKPOINT_HRS	Project HRS Check Point	00 – Management\03 – Check Points

Name Document	Description	File Location
PROJ_CHKPOINT_SDS	Project SDS Check Point	00 – Management\03 – Check Points
PROJ_CHKPOINT_HDS	Project HDS Check Point	00 – Management\03 – Check Points
Requirements		
PROJ_URS	Project URS	10 – User Requirements
PROJ_DEV_DEF	Project use cases	10 – User Requirements
PROJ_URS_VT	Project URS and Validation Test Traceability Matrix	10 – User Requirements
Risks		
PROJ_RSK	Project Risk Analysis	20 – Risks
Specifications		
PROJ_SRS	Project SRS	30 – Specification\00 – Software
PROJ_HRS	Project HRS	30 – Specification\01 – Hardware
PROJ_SRS_FT	Project SRS and Functional Test Traceability Matrix	30 – Specification\00 – Software
PROJ_HRS_FT	Project HRS and Functional Test Traceability Matrix	30 – Specification\01 – Hardware
Design		
PROJ_SDS	Project SDS	40 – Design\00 – Software
PROJ_HDS	Project HDS	40 – Design\01 – Hardware
PROJ_SDS_DEVT	Project SDS and Development Test	40 – Design\00 – Software

Name Document	Description	File Location
	Traceability Matrix	
PROJ_HDS_DEVT	Project HDS and Development Test Traceability Matrix	40 – Design\01 – Hardware
PROJ_DESIGN_FILE_999	Other design documents: diagrams, schematics, mechanical drawings, etc.	40 – Design\00 – Software 40 – Design\01 – Hardware
Verification		
PROJ_SW_FUNCT_VERIFPLAN	Software Functional Verification Plan	50 – Verification\00 – Software
PROJ_SW_FUNCT_VERIFREP	Software Functional Verification Report	50 – Verification\00 – Software
PROJ_HW_FUNCT_VERIFPLAN	Hardware Functional Verification Plan	50 – Verification\01 – Hardware
PROJ_HW_FUNCT_VERIFREP	Hardware Functional Verification Report	50 – Verification\01 – Hardware
PROJ_SW_DEVT_VERIFPLAN	Software Development Verification Plan	50 – Verification\00 – Software
PROJ_SW_DEVT_VERIFREP	Software Development Verification Report	50 – Verification\00 – Software
PROJ_HW_DEVT_VERIFPLAN	Hardware Development Verification Plan	50 – Verification\01 – Hardware

Name Document	Description	File Location
PROJ_HW_DEVT_VERIFREP	Hardware Development Verification Report	50 – Verification\01 – Hardware
Transference		
PROJ_TRANSFER_VERIFPLAN	Project Transference Verification Plan	60 – Transference
PROJ_TRANSFER_VERIFREP	Project Transference Verification Report	60 – Transference
Validation		
PROJ_VALID_TESTPLAN	Project Validation Test Plan	70 – Validation
PROJ_VALID_TESTREP	Project Validation Test Report	70 – Validation
Configuration Management		
PROJ_BASELINE	Project Baseline	80 – Configuration Management
PROJ_COMP_CNF	Project configuration composition	80 – Configuration Management
Change Control		
PROJ_CHANGE_CTRL	Project change control	90 – Change Control

5.5.2 Project Development Plan [PROJ_DEVPLAN]

The **Project Development Plan** should be updated throughout the project as it must accurately describe the processes undertaken at each stage.

Specifically, in phase 3, it will be necessary to introduce **new roles**, such as the team **that will validate the measuring principle**. Usually, this team comprises biologists, chemists, biomedical experts, etc. It must also be considered that there will be a group of **people belonging to the production team** or the EMS specifying the requirements that must be met for the industrialisation of the device. It should also detail who will be the **person who authorises**

the release of the device, usually the product manager or upper management of the company.

This document also needs to plan and detail **two new processes**:

- Validation:

It must be defined **how the design will be validated**. It must be detailed which validations will be carried out within the technical department and which will be carried out with external resources. This validation must contain **all the necessary tests to obtain the CE marking** or approval from other regulatory bodies.

- Product release:

This section will detail **how the release** (Regul. Des. Req. 17), (Regul. Des. Req. 39) of the developed software and hardware **will be documented and performed** (Regul. Des. Req. 39.2).

Reference shall be made to the baseline document [PROJ_BASELINE] to **indicate which version will be released** (Regul. Des. Req. 17.1). Likewise, it must be detailed that the risk document [PROJ_RSK] includes **the accepted residual vulnerabilities** (Regul. Des. Req. 17.2). Additionally, it will be necessary to record that the validation of the device has been carried out according to the plan [PROJ_VALID_TESTPLAN] (Regul. Des. Req. 17.3).

Specifically for software, it will be necessary to ensure that before its release, any **security-related aspects have been resolved** or analysed and categorized as irrelevant or safe (Regul. Des. Req. 39.4).

Suppose validation, functional, or development tests have not been successfully passed. In that case, the client must assess the risk of releasing the product with them. These errors will be recorded in the corresponding verification report (Regul. Des. Req. 39.1).

5.5.3 Validation Plan [PROJ_VALID_TESTPLAN] and Report [PROJ_VALID_TESTREP]

The **validation stage guarantees system performance** (Regul. Des. Req. 38). It has two functions. On the one hand, as in the transfer

stage, it must **ensure that the product manager or client accepts the developed product** (Regul. Des. Req. 28.5). On the other hand, it must include the **planning and record of all those tests required to validate the device and its final certification by the certifying bodies**. Through these tests, all URSs must be fully verified.

The **tests must be carried out by staff not involved in developing the device**; in this way, any conflict of interest can be avoided. Additionally, the correct execution of the tests can be ensured (Regul. Des. Req. 38.2). Therefore, not all tests must always be carried out within the company developing the device. However, the validation plan [PROJ_VALID_TESTREP] **must consider how and where these tests will be passed**. It must also **include evidence of the obtained results** [PROJ_VALID_TESTREP].

In embedded devices, this stage considers **functional safety tests, environmental tests, impacts and vibrations, electrical safety**, etc. (Regul. Des. Req. 7). Also, if applicable, other **tests such as cleaning and disinfection or others must be included**. These tests usually are defined by **IEC 60601 standard family**.

Usability tests, according to IEC 62366, must also be performed, specifically **summative evaluation** tests (Regul. Des. Req. 25); this guarantees that once the design has been completed, its use does not represent a risk.

Tests related to the measuring principle or device's core must also be conducted; **Clinical Evaluation and Clinical Investigation or Performance Evaluation and Performance Studies** are considered in this phase (Regul. Des. Req. 2), (Regul. Des. Req. 4). Depending on the device and its classification, different tests will apply.

Therefore, the validation **tests must guarantee that the development is suitable for industrialisation**. This validation must be carried out in conjunction with the department or company in charge of the manufacturing of the device (Regul. Des. Req. 28.6).

Finally, validations must **guarantee the integrity of the generated software, scripts, executables**, etc. (Regul. Des. Req. 39.3)

All these testing requirements must be identified in the URS document as input to the design. The company's internal or external regulatory department should generate this input.

In the validation stage, three main deliverables are expected to be generated: Product Development Validation Plan [PROJ_VALID_TESTPLAN], URS and Validation Test Traceability Matrix [PROJ_URS_VT], and Product Development Validation Report [PROJ_VALID_TESTREP].

The Product Development Validation Plan document [PROJ_VALID_TESTPLAN] will include the following sections:

- Product Development Validation Test Plan:

The validation plan must define all required tests to validate the development. It should **specify on which devices these tests are to be performed**. Those **devices must be representative of the final product**. It is mandatory to complete the validation using devices **manufactured with the same or equivalent production resources and procedures as those used to manufacture the certified medical device**.

Each validation test will be defined in the following format: VT_XXX_YYY_999, where XXX and YYY are two nouns related to the test to know to which functional block it corresponds. The 999 will be a number used to enumerate the different tests. As in the transfer tests, for the definition of the validation tests, the objective of the test, its description, the configuration, and the acceptance criteria must be detailed. Table 5.5-2 presents the proposed format for defining a Product Development Validation test.

Table 5.5-2: Validation test definition.

VT_XXX_YYY_999: Test name	
Objective	
Description	
Configuration	
Acceptance Criteria	

- Client Transference Plan:

This section will be equivalent to that detailed in the transfer stage. It should include **how the development will be transferred to the client**, detailing the steps to be followed, the timing of the development's acceptance, the deliverables, etc.

- Client Transference Verification Plan:

This section will follow the same approach as in the transfer document. The **project's acceptance criteria will be defined, specifying the tests to be executed by the client**. Likewise, suppose the client or product manager is responsible for coordinating or executing validation tests. In that case, these must be detailed in this section. It will also be necessary to establish a period after which the client will provide evidence of the execution of these validation tests.

The traceability between Validation Tests and verified URS must also be registered and collected in the URS and Validation Test Traceability Matrix [PROJ_URS_VT].

- URS and Validation Test Traceability Matrix:

The **traceability between the specified URS and the defined tests** shall be recorded for the defined tests. This traceability is collected in the Validation Traceability Matrix document [PROJ_URS_VT]. As presented in Table 5.5-3, for each URS detailed with its identifier and name, the test(s) used to validate it must be defined.

Table 5.5-3: URS and Validation Test Traceability Matrix definition.

Identifier	URS Name	Validation Test Correlation
URS_XXX_YYY_999		VT_XXX_YYY_999

Finally, all validation tests must be registered in the corresponding deliverable: Product Development Validation Tests Report [PROJ_VALID_TESTREP].

- Validation Tests Report [PROJ_VALID_TESTREP]

The validation process ends with the **execution of the tests verifying the defined URSs** and the transference of the development to the product manager or client.

As with the transfer tests, the following must be recorded: the test identifier, test name, test objective, date and responsible, versioning tag for software and hardware source code, result and evidence. The format is presented in Table 5.5-4.

Table 5.5-4: Validation test report definition.

VT_XXX_YYY_999: Test name	
Objective	
Date	
Responsible	
Code, hardware tag	
Result	
Evidence	

5.5.4 Medical Product Consolidation Checkpoint

The development ends when the proposed V-model is fully executed. Afterwards, the checkpoint or control gate of phase 3 must be executed.

At this checkpoint, **it will decide if the design and development are successful and whether to proceed to the next phase**. This decision is taken using information and prototypes generated during the design and development phase, as well as other external elements that help to assess the feasibility of commercialising the product. Usually, market information, feedback from potential users, etc., are used as input.

If the **assessment is positive, registering and certifying the developed device with the notified bodies must be done**.

This decision is usually taken by the company's upper management that owns the developed medical device.

6. Methodology Validation

After presenting the methodology, this section aims to validate the proposal. On the one hand, the different use cases in which this methodology has been successfully applied are detailed. For each of them, the phases of the methodology that have been applied are specified. However, most of the use cases are associated with industrial clients, so due to confidentiality reasons, the description of these use cases is done in a general way without going into technical or client-specific details. Nevertheless, the most relevant information is presented to validate the methodology. Likewise, the use case of a wearable, whose development has been carried out following the proposed methodology, is detailed in more depth. Similarly, it is reported that the methodology has been implemented in Tekniker, mainly in the Electronics and Communications Unit, where embedded medical devices are developed. These procedures have been audited during the ISO 13485 certification process. On the other hand, state-of-the-art evidence supporting the proposed methodology is reviewed, with particular emphasis on those studies that validate the different stages of the methodology.

6.1 Successful Methodology Use Cases

This section compiles successful use cases developed completely or partially following this embedded medical product design and development methodology. It also details the deployment of the methodology as a medical device development procedure at the Tekniker research centre.

6.1.1 Use Case 1: Medical Device - Wearable Device Sensor

The wearable solution involves vital signs monitoring platform in the form of a smart wristband, which can collect relevant data about the user's health. Currently, many smartwatches or bracelets monitor vital signs. Still, most of them are not developed to comply with medical device regulations, so the data they provide cannot be considered reliable.

The difficulty in validating and certifying medical devices means that manufacturers do not certify them as such and instead add disclaimers indicating that they are not diagnostic devices.

This development aims to provide a wristband that offers reliable diagnosis and accurate results; this is achievable as the proposed methodology allows us to address in a clear and structured way the technical and regulatory requirements that this type of device has to face. Regarding its category, it is a device **classified by MDR as IIb and IEC 62304 as class B**.

The system uses Bluetooth communication to monitor vital signs and other metrics remotely. In this way, it provides healthcare specialists with enough information to prevent and detect possible illnesses or accidents in patients at home. The device incorporates cutting-edge, low-cost and low-power technology. It is designed to be user-friendly and a fully automated measurement system requiring minimal user interaction.

Phases 1 and 2 of the proposed methodology have been fully implemented in this development. However, **phase 3 is still in progress**. The development is in the validation phase, where **usability tests have been carried out** in a real environment, namely in a nursing home for the elderly. **Clinical trials and the certification process of the device are still pending**.

6.1.1.1 Methodology Phase 1 – Development Feasibility

The implementation of the methodology begins with the execution of phase 1. To this end, firstly, the existing technologies and solutions for these types of devices are reviewed.

In addition, the critical elements are identified: the measurement stage, particularly the measurement of the oxygen, heart rate and the performance of electrocardiograms. After this, the first approximation of the solution is performed to clear the uncertainties of the measurement process.

6.1.1.1.1 Monitoring of vital signs

Monitoring systems aim to obtain continuous patient health information for diagnosis and treatment. These systems can be used in a home environment to characterise the patient status fully. Typically, the indicators measured fare vital signs.

Several patient monitoring solutions can be identified in this context. The most relevant devices are non-intrusive sensing solutions based on highly portable devices and wearables. Currently, technology allows the integration of several sensors, making it possible to build very compact and accurate wearables.

In an ageing society with the increasing prevalence of chronic diseases such as neurological diseases, cardiovascular diseases, diabetes, respiratory disorders, etc., the demand for continuous monitoring will lead to a growing sector of wearable devices. Additionally, it will provide greater patient comfort and more meaningful data to perform diagnostics. As an indication of the financial size of these wearables market in the US, it accounted for spending in the years 2017, 2018 and 2019 of $7 billion, $10 billion and $25 billion, respectively. In other words, spending has almost quadrupled in two years [323].

Vital signs are a series of parameters that show the body's most basic functions, such as a patient's haemodynamic status. They reflect the organism's state and are the first sign of alarm in the event of a malfunction or defect in the organism. There are four main vital signs that physicians and other health professionals

routinely examine in clinical practice [321], [322]: body temperature, heart rate, blood pressure and respiratory rate.

- Body temperature:

Body temperature measures the body's ability to generate and eliminate heat. Three types of temperature can be distinguished depending on where it is measured. Core body temperature, when measured rectally, orally, or tympanically. Proximal skin temperature is measured near the body's central axis, such as the groin or armpit. Furthermore, distal skin temperature is measured in the regions furthest from the body's central axis, typically the hands and legs.

Different results are obtained depending on the place where the measurement is taken. Likewise, aspects such as stress can vary body temperature. In [324], the author states that stress causes a decrease in body and distal temperature but increases the proximal temperature. Likewise, [325] shows how the body temperature presents a circadian rhythm, increasing its value during the day and decreasing during the night.

In addition, age, sex of the patient or different diseases can also affect the measurement. In Alzheimer's patients, the core body temperature rises by up to 0.2 degrees Celsius [326]. A rise in body temperature is the first symptom of infection or inflammation somewhere in the body. When the proximal skin temperature value is below 35.8ºC, it is called hypothermia. If the value is high, it is called febrile (up to 37.5ºC) or fever (above 38ºC) [327].

- Pulse or heart rate:

It refers to the number of heartbeats or contractions per minute. It can change throughout the day or in a given situation. However, it is quickly reversed in the event of a specific triggering situation. The typical values are between 60 and 100 beats per minute (bpm). Tachycardia is defined as a state when the heartbeat is greater than 100 bpm, and bradycardia is when it is less than 60 bpm. The heartbeat is also subjected to natural variations that show how our nervous system adapts to sudden challenges [328].

- Blood pressure:

Blood pressure is the force applied against the walls of the arteries when the heart pumps blood through the body. It is a parameter that can change throughout the day. It is measured in millimetres of mercury (mmHg). Two different blood pressure values can be distinguished, Systolic blood pressure (SBP) and Diastolic blood pressure (DBT).

SBP reflects the pressure in blood vessels when the heart contracts (standard values between 110 and 140 mmHg). DBT is the blood pressure on the arteries' walls when the heart rests between beats (typical values are between 70 and 90 mmHg). A patient is said to be hypertensive when their SBP is above 140 mmHg, and their DBT is above 90 mmHg [329].

- Respiratory rate:

It quantifies the number of breaths taken in a specified period, usually one minute. A standard value is considered between 12 to 20 breaths per minute in adults. When the value is higher, the patient lacks oxygen; this situation is called tachypnoea. If the respiratory rate is lower than the reference value, it is called bradypnea [330].

- Oxygen saturation:

Oxygen saturation reflects the amount of oxygen available in the blood, a critical parameter in patients with respiratory pathology. The standard oxygen saturation value is between 95% and 100%, indicating that cells receive enough oxygen to preserve their function.

A saturation value below 90%, called hypoxaemia, is considered insufficient and is manifested by shortness of breath and a compensatory increase in respiratory rate. Values below 80% are considered severe hypoxaemia [331].

- Blood glucose:

The sugar ingested with food is converted by metabolism into glucose, which travels through the bloodstream to reach cells of different tissue types providing the energy they need to function.

Blood glucose levels, clinically referred to as blood glucose, vary throughout the day. When insulin metabolism is not working properly, glucose is no longer assimilated adequately by tissue cells, accumulating it in the blood. The standard glucose value before eating is between 70-100 mg/dl [332].

6.1.1.1.2 Measurement technologies and principles

This section reviews the specific medical technologies and equipment already established for measuring vital signs. Although there may be variants based on more precise methods, this summary prioritises devices that can measure in an automated non-invasive way.

- Clinical thermometers:

They measure body temperature and can be classified according to the body area for which they are designed or their technology. In terms of technology they use, the most common clinical thermometers are those based on liquid, liquid crystal, electronic contact and infrared.

Liquid thermometers are based on the thermal expansion of a liquid inside a graduated glass tube. The traditional solution used mercury, which is no longer used due to its toxicity. Now, coloured alcohol or gallium is used as an alternative.

They take between 3 and 10 minutes to obtain a reliable measurement and are commonly used for armpit, mouth, or rectum measurements. Due to their fragility, measurement time and the ban on mercury variants, digital alternatives have largely displaced their use, especially outside the hospital environment.

In addition, there are liquid crystal thermometers consisting of heat-sensitive liquid crystals integrated into a plastic strip. These crystals change their colour to indicate different temperatures. They are usually placed on the forehead and are disposable [333].

Electronic contact thermometers are made up of temperature-dependent transducers that vary the output voltage depending on the patient's temperature. This voltage variation is translated into degrees and displayed on a small screen.

Regarding advantages, they are easy to read and quick to respond to. For this reason, their use has spread inside and outside the hospital setting. Many employ predictive algorithms to provide a reading in a few seconds rather than a minute. They are commonly used to measure the armpit, mouth, rectum, or ear. In the case of predictive devices, the algorithms must consider the placement area to provide an accurate temperature reading [334].

Infrared thermometers do not require physical contact to carry out the measurement, and it is usually performed on the forehead. However, there are specific developments for ear measurements. As they do not require contact, they can reduce the risk of infection transmission. Measurement times are low, in the order of seconds.

As these devices are based on optical sensors, readings can be affected by the state of the surface on which the measurement is made (cleanliness, humidity, position, movement, etc.). Also, the measurement can vary due to ambient light level, external heat sources or the use of clothing or cosmetic products. Therefore, they are more prone to measurement errors than contact alternatives [335].

- Heart rate monitor:

A heart rate monitor allows real-time measurement of a patient's heart rate. Modern wearable devices typically use electrocardiography or photoplethysmography methods to record heart rate signals.

Electrocardiography (ECG) can record the electrical activity of the heart. That is, the bio-potential generated by the electrical signals that control the expansion and contraction of the heart. In other words, it captures, records, and magnifies the heart's electrical activity. Different types of ECGs can be distinguished according to the number of used electrodes: 1, 2, 6 or 12 leads or channels, where each lead will measure the electrical potential difference between two electrodes. A 1-lead ECG provides only essential monitoring of the heart. In contrast, a 12-lead ECG provides a complete picture of cardiac activity. ECG is widely used to detect almost any cardiac pathology [336].

There are different algorithms to extract the pulse from the ECG. The basis of this measurement is the detection of the QRS complex. This parameter is formed by three vectors: the Q wave, the first wave of the complex with negative values; the R wave, which follows the Q wave, is positive and the largest; and finally, the S wave, any negative wave that follows the R wave. Based on the duration, amplitude, and shape of the QRS complex, it is possible to detect heart rate, arrhythmia, infarcts, and other disorders. In [337], authors discuss several algorithms to estimate heart rate frequency based on auto-correlation, thresholding and peak detection in signal energy envelope.

Photoplethysmography (PPG) determines the heart rate using a light source of a specific length that emits a beam on the skin to illuminate the subcutaneous vessels. The subcutaneous vessels reflect part of the beam depending on the number of red blood cells they contain [338]. The reflected light hits on a photosensor, converting it into an equivalent voltage. The cardiac cycle can be obtained by measuring the interval between each voltage peak. Its principle of operation is the same as that of the oximeters; hence, pulse-oximeters offer both measurements.

- Blood pressure monitor:

It is a medical device that indirectly measures blood pressure through SBP and DBT values. It consists of a manometer and an inflated cuff which squeezes the measurement area so that, by occlusion, blood transit is temporarily stopped for measurement. It is used with a stethoscope to auscultate the audible intervals of the Korotkoff arterial sounds while the cuff is being deflated in a controlled manner.

The digital ones are the most appropriate for home use as the whole process is automatic, including inflation. However, they require periodic calibration as they use sensors placed in the cuff to detect Korotkoff sounds [339]. These sensors can be auscultatory or oscillometric.

The auscultation is based on microphones capable of interpreting the sounds of the measurement process [340]. The Oscillometer relies on deformable membranes whose variation in piezo-resistance

or capacitance allows the analysis of the vibration transmission of the arterial wall [341].

- Respiratory rate monitor:

Respiratory rate is usually measured manually by observation, palpation or using a stethoscope. However, there is equipment for automatic monitoring. Respiratory rate can be measured using four methods, respiratory inductance plethysmography, impedance pneumography, spirometry and capnography.

Respiratory inductance plethysmography is based on a device that records respiratory movements using an inflatable coil surrounding the thorax. These signals are connected to a monitoring device that transforms the inductance of these coils into signals relative to the rib cable and abdomen strains [343]. This measurement technique is widely used in the hospital environment.

Impedance pneumography is based on impedance generated by the chest while breathing. This measurement is done using 2 or 4 electrodes on the thorax. A high-frequency, low-amplitude current flows through the chest cavity and the variation in resistance is used to estimate respiratory rate [344]. The variation in resistance is due to body impedance and its respiratory cycle.

By using spirometers, it is possible to record the amount of inhaled and exhaled air during a specific time. Modern spirometers can graphically represent these curves. Based on a test of at least 60 seconds, it is possible to measure the breathing rate. For that, it is enough to count the number of peaks or troughs represented on the breathing graph [345].

Capnography measures the concentration of carbon dioxide in a patient's airway during the respiratory cycle. The respiratory rate can be determined from the time evolution of this concentration. Their operation is based on carbon dioxide's absorption principle of infrared light [346].

- Pulse Oximeter:

This medical device can determine in a non-intrusive manner the percentage of haemoglobin’s oxygen saturation using photoelectric

methods. The pulse oximeter is placed on a relatively translucent part of the body with good blood flow: the fingers, toes, earlobe, or wrist. The equipment emits light at specific wavelengths (green/red/infrared), which pass sequentially from an emitter to a photodetector through the patient [347]. The absorbance of each wavelength caused by arterial blood (pulsatile component) is measured, excluding venous blood, skin, bone, muscle, and fat. With this data, it is possible to calculate blood oxygen saturation.

- Glucometer:

It is a device in which a test strip is impregnated with a blood drop. It provides the result of the patient's blood glucose levels in just a few seconds, and its use is not complex for the patient himself. However, it is an invasive method [348]. As an alternative, in the article [349], developments that aim to achieve continuous monitoring of glucose concentration, usually in interstitial or tissue fluid, are presented.

6.1.1.1.3 Multi-monitoring solutions

The current scenario of technology applied to health services presents challenges when considering the possibility of using monitoring devices at home. The solutions are evolving and making remarkable advances, such as developments integrating micro and nanoelectronics. As these components are programmable and have storage capacity, they are the basis for creating small systems with integrated sensors and information processing capacity. Additionally, the evolution in embedded communications makes it possible to connect these systems with remote information repositories or even to be interconnected.

This scenario is perfectly compatible with introducing and using highly portable, increasingly usable and interconnected health and personal care devices. These principles tie in with concepts that are at the epicentre of the evolution: eHealth, Internet of Things (IoT) and its combination IomT (Internet of Medical Things) [350].

Thus, advances in microelectronics, communications, sensors and data processing have made possible a great scope in developing new technologies and devices to support healthcare. Wearable sensor devices for monitoring are an example [351]. There are a large

number of wearable devices that, due to their design and built-in sensors, are functionally adapted to obtain health-related information from non-hospital environments. They can continuously monitor parameters such as body temperature, position or bio-electrical signals [352].

However, developing wearable devices with high usability and medical use still presents several challenges. Despite significant developments, it is still impossible to bring together the ability to monitor all the parameters for comprehensive care in a single device. The proper sensor placement is critical for correctly measuring vital signs.

Various systems are proposed in the literature for monitoring patients' vital signs at home [353], [354]. However, many of these monitoring systems are designed only to monitor one or two specific parameters [355]. Among the multi-measurement systems, the most important ones include smartwatch-based systems, smart furniture, and textiles with integrated sensors.

- Smartwatches:

Smartwatches play an essential role when monitoring vital signs. Some devices offer quite advanced health parameter monitoring, including ECG-correlated measurements. However, it is required to certify these smartwatches to use them as medical devices. The regulatory barrier limits their availability in certain markets.

A representative example is the Apple Watch, the first smartwatch approved by the US Food and Drug Administration (FDA) and which incorporates algorithms for detecting atrial fibrillation and performing ECGs [356]. A similar authorisation was granted by the European Commission in 2019 for 19 countries. For this purpose, the watch incorporates electrodes on its back and crown. Placing a finger on the crown, the user closes the circuit with the back electrodes providing data to the ECG application. However, this device has limitations as it is a single-lead system that cannot detect heart attacks, cardiovascular accidents, or other heart conditions.

Regarding other vital signs, such as blood pressure measurement, the technology available for wearables is still far from healthcare-related purposes. The Omron Heartguide wristband [358] is the only

wearable with FDA clearance. This device is an ultra-portable wrist sphygmomanometer.

Likewise, there are more and more references in which different smartbands or smartwatches are used to estimate body temperature. For example, [359] presents a statistical approach to estimate body temperature based on skin temperature measured with a smartband.

- Smart furniture:

Smart beds or chairs integrating vital signs sensors can also be an interesting option for non-intrusive monitoring of vital signs.

In [360], a set of pneumatic sensors placed on a bed is presented to measure heart and respiratory rates. In [361], temperature measurement is carried out using a high-precision IR camera. Authors in [362] use external sensors to measure respiratory and heart rates in bed by integrating accelerometers in the blanket.

- Textiles with sensors

Another alternative is incorporating monitoring capabilities into patients' everyday items or accessories, such as socks, shoes, T-shirts or waistcoats [363]. One example is the Smart Vest, a wearable monitoring system for heart rate, blood pressure, axillary temperature and ECG [364]. Other experimental designs with promising preliminary results incorporate heart and respiratory rate measurement capabilities into conventional T-shirts [365].

Article [366] describes the pilot development of a commercial home telemonitoring system. The system can perform ECG measurements (1-lead) using an intelligent patch placed on the chest or integrated into a garment or an elastic band. The system also employs a commercial multi-purpose non-wearable device to measure several constants such as heart rate, blood pressure, and oxygen saturation. The same system also performs a glycaemia measurement.

In [367], an intelligent garment with textile electrodes is used to measure -ECG and a chest strap to measure respiration rate using an inductive transducer that measures est or abdominal circumference changes.

An example of a commercial development is the Hexoskin smart garment [368]. It is an elastic T-shirt that provides continuous information on heart and lung activity. It provides ECG (1-lead), heart rate, heart rhythm and breathing rate. Another example is the Philips wearable biosensor [369]; in the form of a patch, it includes a temperature sensor and 1-lead ECG.

6.1.1.1.4 Technical feasibility analysis

The challenge of measuring vital signs means that the first phase focuses on validating the feasibility of two of the most critical measuring processes, heart rate and oxygen. Photoplethysmography (PPG) and Electrocardiography (ECG) are proposed to measure these parameters. PPG enables measuring both oxygen and heart rate as if the vessels are illuminated; the reflected light is proportional to the oxygen and heart rate values. Using the ECG, recording the heart's signals is possible, for which electrodes are necessary.

Measurement feasibility verification is done with a first prototype, including the sensing stage. In addition, as the housing is particularly relevant for light emission and ECG performance, a 3D-printed version of the device's housing is used to validate the concept. Figure 6.1-1 shows the first prototype.

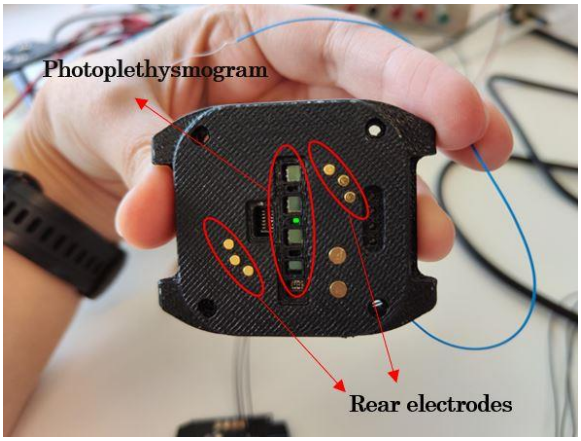


Figure 6.1-1: First prototype to validate feasibility.

The obtained initial results were not satisfactory. On the one hand, for the PPG, when the intensity of the emitted light was high, the

housing could not isolate the LED channels from the photodiodes. Therefore, the need for spacers between the photodiodes was identified to further isolate the LED channels and photodiodes. In addition, the received signal was too weak as the housing was too thick, making it impossible for the photodiodes to come into direct contact with the skin. For all these reasons, it was decided to use a Teflon sheet to simulate the isolation between LED channels in the final housing. A provisional PCB assembly was then made (see Figure 6.1-2). After testing, a good signal from the PPG was present; moreover, it was concluded that using two LED/photodiode pairs was enough.

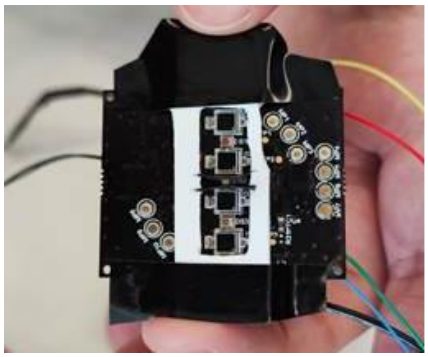


Figure 6.1-2: Assembly to validate the feasibility of the PPG measurement.

Regarding ECG measurement, although the design allowed obtaining the signal from the heart, it was too weak. It was concluded that this was due to the area of the electrodes. This hypothesis was validated by directly welding larger electrodes. The results obtained were satisfactory. Figure 6.1-3 shows the unfiltered PPG and ECG values. These values were obtained directly from the sensing stage without filtering the signal. Once the project and product managers have validated the results, phase 2 of the development proceeds.

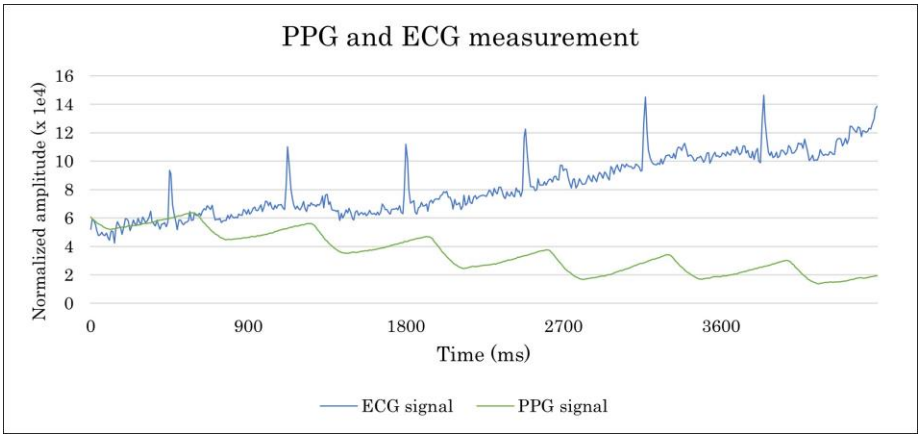


Figure 6.1-3: Measurements made with PPG and ECG sensor.

6.1.1.2 Methodology Phase 2 – Incremental and Iterative Prototyping

Functional development of the device is carried out incrementally and iteratively following phase 2 of the proposed methodology. In the following chapters, the development is presented.

6.1.1.2.1 Detailed description of the final solution

A smart bracelet is designed and developed to remotely monitor the vital signs of elderly or dependent patients at home.

This device is equipped with embedded electronics to perform photoplethysmography to calculate heart rate and blood oxygen saturation and 1-lead electrocardiography between the right and left hand. Additionally, it can measure body temperature and physical activity.

Measurements are regularly taken with minimal patient intervention, making it as simple as possible. Patients place their right hand on the device when prompted by the wearable to perform the electrocardiogram. The data is stored in the internal memory and can be downloaded through USB using the charging platform or Bluetooth. Table 6.1-1 lists the main wearable's technical characteristics.

Table 6.1-1: Wearable's technical specification.

User interface	RGB Led and Buzzer
Communication interface	USB and Bluetooth 5.1 (BLE)
Battery	370 mAh (1 week)
Internal flash memory	4 Gb
Size	49 mm x 45 mm x 15 mm
Electrocardiogram Interface	Two rear electrodes and two front electrodes
Photoplethysmography interface	One red LED, one infrared LED, one green LED and two photodiodes

6.1.1.2.2 Final Design

The device's electronics are embedded in a square sphere and are divided into two connected PCBs; one is used for the sensing stage, and the other for the control stage. Both PCBs are held in the lower case, while the battery is in the upper case. A conventional, adjustable silicone strap is fastened to the wrist. As presented in Figure 6.1-4, the LED indicator and the front electrodes are located on the front of the device.

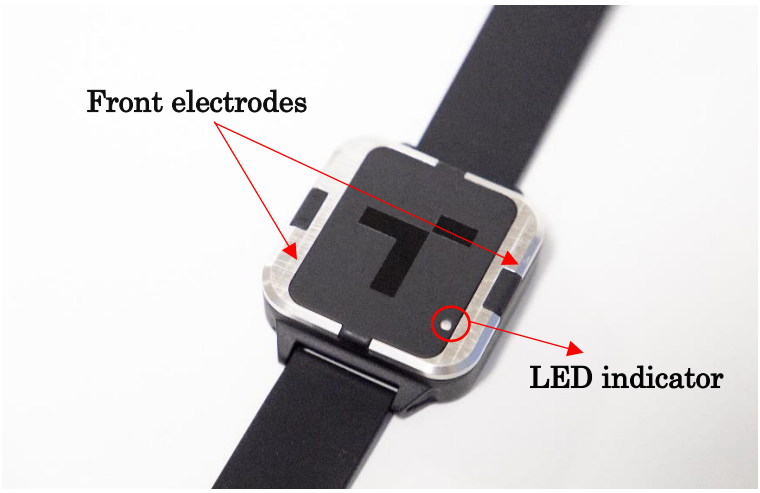


Figure 6.1-4: Front side of the device.

Additionally, on the back of the device are the rear electrodes and LEDs and photodiodes used for photoplethysmography. The

charging and serial communication port is also on the wristband's back. Figure 6.1-5 shows the location of these elements.

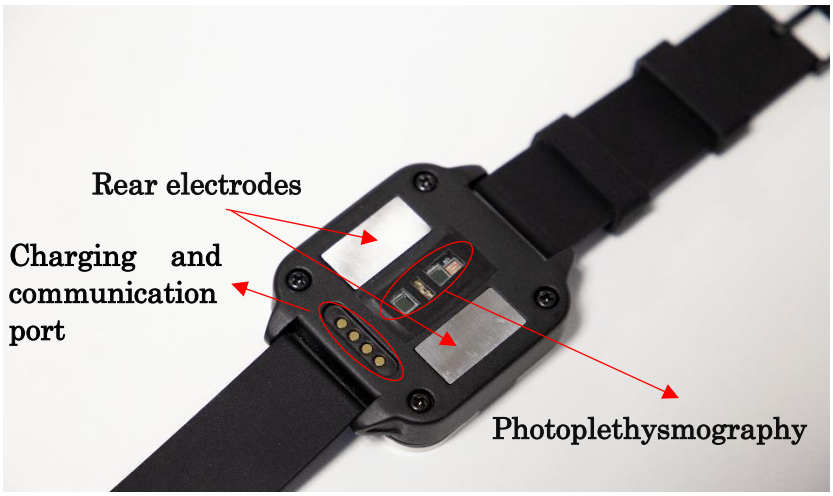


Figure 6.1-5: Rear side of the device.

The sphere size is 49 mm x 45 mm x 15 mm. Figure 6.1-6 presents these dimensions graphically.

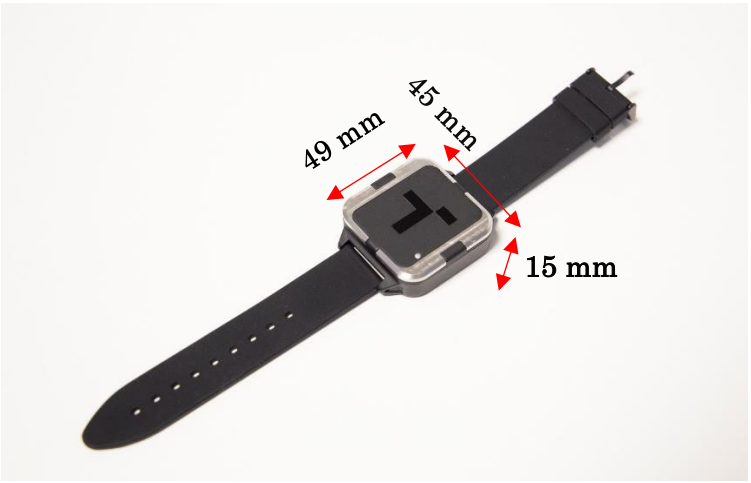


Figure 6.1-6: Device sizes

6.1.1.2.3 Use mode

The wristband must be placed on the patient's non-dominant wrist. The most appropriate height to place it is slightly above the wrist

bone to avoid the area with the highest concentration of bones and ligaments. As seen in Figure 6.1-7, this position must be aligned with the centre of the device's back face. The correct placement of the device is critical for PPG data collection. The smartband should be attached so that the device does not move with the movement of the arm.

Figure 6.1-7: Device placed on the wrist.

Twice a day, an electrocardiogram will be performed on the patient; the beginning of the process will be indicated by the green LED and a series of three beeps. A time window of 10 seconds shall be given for the patient to place the fingers or hand palm on the front electrodes. The green LED shall remain on for the entire electrocardiography process, which shall be at around 30 seconds. After this time, the end of the measurement will be notified with three beeps, and the green LED will turn off. It is recommended that the patient rests the fingers or hand palm on the front surface and presses lightly to obtain a clearer ECG signal to perform the test.

The wristband is also programmed to take a PPG measurement every hour. The signal quality will vary depending on the patient's current activity. The patient will not be alerted when this measurement is taken.

Other data, such as patient temperature and instantaneous acceleration, are acquired periodically without indication.

When the device's battery charge drops below 20%, the LED indicator will turn red, indicating that the wristband needs to be

charged. The charging of the device is done using the charging station; the smartband must be placed on it, aligning the terminals of both devices. The LED indicator will turn blue while the device is charging. When the charging process is complete, the LED indicator will turn green. Figure 6.1-8 shows how to place the device in the charging station.

Figure 6.1-8: Wristband on the charging station.

6.1.1.3 Methodology Phase 3 – Medical Product Consolidation

Finally, phase 3 is executed; in this phase, the presented V-model is fully executed, consolidating all the development, verifying each component, and generating the required documentation.

The device has been tested to verify the reliability of the measurement results. Firstly, temperature tests in a climatic chamber have determined the temperature sensor's accuracy. Regarding the SpO2 measurement, a signal-to-noise ratio measurement was performed by placing the device on the charging station; this is possible as this dock blocks external light artefacts and reflects the LED signal. In addition, both SpO2 and ECG measurements have been verified using a calibrated measuring device. Angular speed and linear acceleration were tested by driving the device on the axis of a controlled motor. Finally, the inclination has been checked against a calibrated measuring device.

Likewise, environmental and functional safety pre-certification tests have been carried out with satisfactory results.

The development is currently in the validation phase. On the one hand, usability tests have been carried out in an assisted living facility. On the other hand, final environmental, electrical safety, and clinical trials are still pending.

6.1.2 Use Case 2: Medical Device for Professional Use

This use case involves the use of the methodology in developing a **medical device for professional use**. The client and owner of the development is a **start-up company** that does not have enough technical expertise to develop the product entirely. Therefore, they decided to outsource part of it.

The client's contribution to the development is strictly limited to the definition of the algorithm that is the basis of the detection performed by the device.

Specifically, it is an embedded device, including embedded software and hardware. It is intended to be sold in Europe and is **classified by the MDR as Class IIa**. The software can be classified **as class B, according to IEC 62304**.

During this development, **phases 2 and 3 of the methodology have been applied**. Phase 1 was not executed because the client already had a preliminary prototype with which the feasibility of the development was validated.

The application of the proposed methodology can be considered a success, as this product is currently in the process of CE marking and has already **successfully passed all the validation tests, including electrical safety tests, environmental tests and clinical trials on patients**. The design is expected to be validated and CE-marked by the competent authorities in the coming months.

6.1.3 Use Case 3: In-Vitro Medical Device for Professional Use

In this use case, **the methodology has been applied in its entirety**. During the first phase, the main feasibility concerns and the selection of critical components were resolved. The incremental and iterative development of all the device's functionalities was carried out in the second phase. Finally, during the third phase, the product

was consolidated through an intense documentation and verification process.

The **company** that owns the product is **well established** and has been commercialising products in Europe for several years. However, it does not have the resources to design and develop the device, so it outsources this activity. This device has been designed and developed for **professional use** in the European market. It is **classified as class B by the IVDR and class C by IEC 62304**.

The development has required the design and development of embedded software and hardware. Currently, the **product is CE marked** and sold in Europe for some months. During the environmental, safety and clinical testing phase, as well as the device certification, no problems were found with the work completed during the design and development phases. The client was also satisfied with the developed product and the management during the design and development phases. Therefore, it can be concluded that **this product has allowed the complete validation of the proposed methodology**.

6.1.4 Use Case 4: In-Vitro Medical Device for Home and Professional Use

In this case, the **three phases of the methodology are applied**. In contrast to the previous use case, it was not necessary to develop the entire device but rather to extend the functionality of an existing device.

The company, a **consolidated entity** in the healthcare market, does not have a technical department, so it decided to outsource this development. Specifically, the development includes the design and development of embedded software and hardware for a **home and professional use** device, which the **IVDR classified as class B** and **IEC 62304 as class C**.

This development is on the **market in Europe and the United States**, and no problems have been identified with applying the proposed methodology. Thanks to this use case, the methodology is validated not only for the integral design of new medical products. It is also **used to evolve or extend the functionality of devices already on the market**. Furthermore, even though this methodology aims to

address European regulatory needs, the **FDA and European Commission have approved this device** following the proposed methodology.

6.1.5 Use Case 5: In-Vitro Medical Device for Professional Use

The **entire methodology** has been applied in this development. This device is for **professional use** and is regulated by the European **IVDR** regulations as *class B* and **IEC 62304 as class B**. The client, a **consolidated company**, has a technical department with embedded software and hardware development capabilities. However, it has decided to outsource this activity due to a lack of available resources.

The development has been carried out following the proposed methodology. A first phase of feasibility verification of the measurement principle was carried out, followed by the functionality development and consolidation process. During this last phase, the client actively participated in the validation of the device, assuming almost all of its execution as its own.

The device is **now CE-marked** and sold in Europe without any problems. The client is also delighted with the result obtained during the outsourcing process.

6.1.6 Other Use Cases

In addition to the use cases mentioned above, this methodology has been partially applied in developing medical devices for several start-ups.

In many cases, the application of the methodology has been limited to the execution of **phase 1, Development Feasibility, and phase 2, Incremental and Iterative Prototyping**. In many cases, executing the product consolidation phase has not been possible. These start-ups still need to obtain enough funding and structure to undertake the product consolidation phase and its subsequent commercialisation.

In the coming months, several of these companies are expected to continue with the development process and undertake the product consolidation phase to homologate and commercialise their devices.

6.2 Certification of the Methodology

In 2021, Tekniker obtained the ISO 13485 certification for designing and developing medical devices. For this, phase 3 of this methodology has been fully implemented throughout the design and development procedures of embedded electronic products carried out in the Electronics and Communications Unit.

The procedures referring to phase 3 of **this methodology have been audited by the company SGS, thus certifying the suitability of this methodology for the design and development of embedded medical devices**. In addition, both in January 2022 and 2023, the application of these **procedures in several medical device developments has been reviewed by SGS**. The outcome of the review has been successful.

6.3 Studies Supporting the Proposed Methodology

The objective of this section is to support the main concepts of this methodology by using studies available on state of the art. Therefore, it is intended to **justify** the **main principles** of this methodology: (i) the **need for the feasibility phase**, (ii) the use of the **Agile** approach for **iterative and incremental development based on prototypes**, and (iii) the use of the **V-model** during the **medical product consolidation** phase.

6.3.1 Feasibility phase

Today, start-ups are the core of innovation in the healthcare sector. However, as discussed in [370], many fail due to a lack of experience in business, medical regulation, and product development. It is for this reason that the feasibility phase takes on relevance. In [370], considering the different challenges faced by start-ups to access the healthcare market, it becomes clear that even during the business valuation phase, it is necessary to analyse the start-up's maturity and its idea. For this reason, the authors present a tool for measuring healthcare start-ups' maturity. The results of this study show the lowest scores in design and production capabilities, regulatory and clinical readiness, and human resources and role evolution.

Not only the feasibility phase is relevant for business-related aspects, but it is also crucial for technical development. In [371], a study is presented on how experimentation in software start-ups can lower the risk of development. According to the study, many of the presented solutions fail due to not identifying the technical challenges that may be involved early in development. The study also shows that start-ups spend much time on development without validating the critical aspects of their developments, which are vital to the success of their solution. The author remarks that after this study, the participating companies have realized that they should invest the time at the beginning of their developments to validate the most critical aspects of their solution.

Likewise, the execution of experimentation or concept feasibility studies is contemplated in the Lean Startup methodology [372], [373]. The basis of this methodology is the execution of MVPs to obtain continuous feedback during the development phase. This consolidated methodology has demonstrated the relevance of this phase in start-up companies. Furthermore, some studies conclude that this methodology can also be used in established companies, obtaining good results [374], [375].

6.3.2 Agile methodology for iterative and incremental prototyping

The proposed methodology uses an Agile approach for iterative and incremental development of the new product's functionality. Furthermore, it suggests using prototypes to obtain feedback from the user or client early in the development process. It also encourages continuous feedback during development through prototyping.

Many studies support using Agile versus Heavyweight methodologies to develop different product types. In [376], the author presents a study among professionals using Heavyweight and Agile for software development. Specifically, the study is based on 87 surveys conducted among professionals from different sectors. The research concludes that Agile and Heavyweight professionals are satisfied with their development processes. However, they agree that the agile methodology improves communication with the client and increases productivity. Furthermore, this study finds that the

quality of work is guaranteed more effectively in Heavyweight methodologies and that there are also fewer role clarity issues. According to this study, it is mainly positive to use Agile during the product's functionality development phase and a subsequent stage of Heavyweight to guarantee product quality.

Although Agile was first used in software development, it has been extended to developing other products and solutions. In [377], the authors study how Agile affects project success. Using data from more than 1000 projects, they analyse two aspects: efficiency and customer satisfaction. The study results show that the use of Agile has a positive impact on both aspects.

In addition, the long-term acceptance of the methodologies is essential. While Heavyweight has been used in organisations for many years, Agile is relatively new. Despite this, some studies support its acceptance in companies that have applied for several years. In [378], an investigation is presented on how developers perceive Scrum in the long term. Specifically, it reveals that developers see Scrum as a framework compatible with their practices and advantageous to traditional methodologies. They also point out that a certain discipline is required to be able to execute projects under Scrum.

Scrum and Kanban stand out among the existing frameworks to implement the agile methodology. In [379], the authors carry out a statistical analysis to compare both frameworks in several factors: budget management, risk management, project quality, resource management, scope clarity and schedule management. The research concludes that there are no statistically significant differences between the two frameworks but that Kanban works better for certain aspects, such as project management. The report finds that both approaches lead to successful project outcomes. Because both approaches provide positive features, this methodology combines both Scrum and Kanban.

Likewise, some studies identify the requirements definition phase as key in projects that follow Agile methodology. In [380], the author conducts an empirical analysis that supports this premise. This analysis has been conducted with 108 participants from several

companies with different natures that use the agile methodology in their development.

Therefore, the proposed methodology also focuses on defining precise requirements before their implementation or execution. Using URSs as the input of the Agile methodology and the basis for defining and planning each iteration is proposed.

6.3.3 V-model methodology for medical product consolidation

The last phase of the methodology is the consolidation of the medical product. For this purpose, it is proposed to follow the V-model.

First, this is because, as presented during the V-model analysis, although medical device standards do not require a particular methodology for device design, these standards show using the V-model as an example or recommendation. This approach is stated in IEC 60601, a standard that aims to guarantee the safety of the medical device, and in IEC 62304, a standard that defines the software lifecycle processes in medical devices.

Some studies support this decision; for example, in the study conducted in [376], traditional Heavyweight methodologies guarantee high quality in new product development.

Likewise, in [381], the author reviews the advantages and disadvantages of the agile methodology versus the V-model for medical product development. This article concludes that the V-model is widely accepted in the medical sector and is suitable for certifying medical products while guaranteeing the system's safety. However, it is stated that it makes development progress more complicated since it does not allow any iteration or feedback from the client during the early stages. In contrast, the Agile methodology eases development. However, it can hinder traceability and the generation of the product documentation required by the regulatory authorities.

The authors in [382] analyse the barriers to applying Agile in the medical device industry. To do so, they surveyed several companies in the medical sector. The survey reveals that 75% of the respondents use Waterfall or V-model methodology, while the remaining 25% use Agile methodology. Among the reasons why

respondents found it challenging to adopt Agile methodologies include: "Lack of Documentation", "Regulatory Compliance", "Lack of Up-Front planning", and "Insufficient coverage of risk management activities".

Therefore, the proposed methodology tries to obtain the best of both methodologies by using the Agile methodology in the development phase and V-model in the product consolidation phase.

7. Conclusions and Future Works

This section reviews the main conclusions of this work. It is concluded that the proposed methodology successfully addresses one of the stoppers identified in the medical device sector: the failure of start-ups trying to design and develop innovative medical products. The main contribution of this work is also detailed: a methodology for start-ups that want to design and develop embedded medical devices under the new strict European medical device regulations. Other contributions made throughout this work are also identified. Likewise, the scientific contributions made in international journals and conferences are detailed. Finally, future lines of research are identified to continue the work carried out in this thesis. On the one hand, extending the methodology by integrating the product manufacturing phase is considered. Likewise, more and more products include technologies such as Artificial Intelligence, so extending the methodology for new solutions is also proposed. On the other hand, during the analysis carried out for medical devices, certain technical and regulatory similarities with other sectors, such as the automotive sector, were identified. Therefore, it is proposed to migrate this methodology to other sectors that are constantly evolving.

7.1 Conclusions

This thesis presents a **new methodology for designing and developing medical devices for start-ups**. This methodology aims to support start-ups to design and develop medical devices **in compliance with the standards** that regulate the sector. The proposed methodology also addresses the needs of established companies facing difficulties adapting their design and development processes to the new MDR and IVDR regulations.

The methodology also **answers the healthcare sector's main challenges**: the need for **constant innovation**, the use of embedded electronics to **develop intelligent and connected solutions**, and the **increasing number of start-ups that fail** on the path to innovation.

The research work presented in this thesis provides the technological and regulatory aspects to be considered for introducing embedded systems for healthcare. This work shows that the healthcare sector requires more and more devices to help healthcare professionals diagnose and treat patients. Developing these devices is not an easy task as these are regulated under strict regulations (Section 1.1).

Therefore, there is an increasing need to develop new medical devices with sophisticated technical solutions that allow the healthcare sector to evolve. The emerging technologies have enabled the increase of start-ups that aim to introduce new devices into the market, with no need to invest significant resources in their development (Section 1.2). Among the technologies that stand out for rapid innovation and relatively low development costs is embedded electronics technology (Section 1.3).

The successful development of new medical devices can be complex, depending on the company's nature. Although the concept of innovation is well understood, the lack of awareness of the relevance of analysis, design and verification phases often causes the death of many medical device concepts along the market path (Section 0).

Developing new medical devices based on embedded systems offers many opportunities for the health sector. However, it is necessary to consider the peculiarities of these systems to build a medical device successfully (Chapter 2).

Additionally, this work shows that new product development in the medical sector requires the knowledge to identify applicable requirements and regulations and establish a quality management system (ISO 13485) to ensure quality during the whole product life cycle (Chapter 3).

Once the technical and regulatory aspects of medical devices with embedded systems have been broadly defined, the need for a methodology to address the sector's requirements is identified. On the one hand, this methodology must contemplate technical, regulatory and methodological requirements. On the other hand, it must minimise the development risk and maximise investment efficiency. Finally, it must be suitable for companies with no experience in the sector.

This work highlights that the high number of applicable standards requires a methodology that eases compliance in an efficient and orderly way. It also shows that the current methods are unsuitable for the sector's needs, nor do they help the nature of start-up companies (Chapter 4).

For these reasons, a new methodology based on three stages has been introduced and validated (Chapter 5). The first stage, Development Feasibility, aims to turn new product ideas into technical solutions, thus minimising the technical and economic risk of the solution.

The second phase, Incremental and Iterative Prototyping, is based on the Agile development of functional prototypes iteratively and incrementally. In this phase, the main documentation and procedures of the medical device are established. However, it avoids traditional and rigid medical product development methodologies that do not allow the introduction of changes and obtaining feedback from the client in the early stages of development.

Finally, in the third stage, Medical Product Consolidation, development consolidation is proposed through a V-model that meets all the regulatory requirements for embedded medical device design and development.

The proposed methodology is particularly interesting for start-ups due to the type of solution they usually have, innovative ideas but

with high technical and economic uncertainties. Consequently, using this methodology, it is possible to meet customers' product requirements by providing a device that complies with European medical regulations.

Finally, this methodology is validated through the presentation of use cases in which its application has been successful and through the analysis of studies that support the presented approach (Chapter 6).

7.2 Principal Contributions

The **main contribution** of this thesis is the **definition and validation of a design and development methodology for medical devices that comply with industry regulations** and is intended to serve as a guide for start-ups and engineering teams seeking to address this type of development. This main contribution is based on research and contributions in three areas:

- Medical product design and development methodologies according to medical regulations:

The biotechnology sector is identified as a field in which the development of innovative devices is in continuous demand. Therefore, this thesis proposes a methodology that complies with the regulatory requirements of the health sector.

Based on the definition of a typical embedded device, the applicable regulations for these devices are analysed. This analysis shows that the existing high number of standards requires following a design and development methodology that eases this process. After analysing the available methodologies, no one covering regulatory requirements was identified.

Therefore, a three-step methodology that addresses medical devices' technical and regulatory requirements is proposed.

Finally, this methodology is validated through its application in developing embedded medical devices. Several of the developments carried out have obtained the CE mark, so it can be concluded that this methodology has been successful. Likewise, the procedures

associated with this methodology have been validated by SGS, ISO 13485 certifying body, during its audit in Tekniker.

- Health product design methodologies applicable to start-up companies:

Aware of the opportunity in the healthcare sector, more and more start-ups are committed to developing medical devices. Despite this, a large number of them end up failing.

The performed analysis identified three reasons why start-ups fail in the healthcare sector: (i) lack of knowledge of the regulations that apply to the sector, (ii) lack of knowledge of market or customer needs, and (iii) lack of financial resources.

Therefore, this thesis proposes a methodology that addresses these challenges and guides start-ups through the design and development processes.

The analysis of methodologies carried out shows that the methodologies usually used by start-ups tend to be Agile methodologies that allow rapid and cheap developments, explicitly seeking a marketable solution as soon as possible.

These methodologies are not always aligned with the regulatory requirements of the health sector as they focus on developing solutions or concepts quickly and cheaply, whereas in the case of medical devices, risk management and device quality are more important.

Therefore, this thesis presents a methodology for the development of medical devices focused on start-ups that aim to address the main challenges they face and minimise the number of start-ups that fail in the sector through a methodology that allows: (i) compliance with regulatory requirements, (ii) early identification and fulfilment of market and customer needs and (iii) minimisation of economic investment in stages with a high degree of uncertainty.

Finally, the proposed methodology has been validated through its application in developing start-up medical devices; some of these developments are already on the market.

- Design methodologies for embedded systems:

The research carried out in this thesis has focused on investigating, defining and developing a methodology that addresses the design and development needs of devices based on embedded systems.

To this end, the typical embedded device is defined, specifying the main elements an embedded product usually includes. All components and their technologies are reviewed to determine a methodology that meets the technical requirements of these developments.

Following this analysis, it becomes clear that an embedded device design, development and validation involve many steps and that a methodology can help make the development more efficient. An analysis of available methodologies in state of the art is also carried out, but no one is identified that explicitly addresses the embedded systems' technical requirements.

Therefore, a methodology is proposed considering all embedded technologies (software and hardware) during all design and development phases. Specifically, phases 1 and 2 of the proposed approach are enough to cover the development of embedded devices that are not for the health sector.

Finally, through the development of embedded products, it has been validated that the methodology works successfully during the design and development processes. Furthermore, these procedures have been established in Tekniker's Electronics and Communications Unit, where embedded devices are currently being developed for several sectors.

7.3 Thesis Articles

This section compiles the different conference papers and articles made throughout the process of definition and validation of the methodology proposed in this thesis, Medical Devices with Embedded Electronics: Design and Development Methodology for Start-ups.

7.3.1 Embedded Sensor Systems in Medical Devices: Requisites and Challenges Ahead

Journal Paper
Date of publication: 16 December 2022
Sensors MDPI (ISSN 1424-8220), Section Biomedical Sensors
Special Issue Embedded Sensor Systems for Health
Impact Factor 2021: 3.847
CiteScore 2021: 6.4
SJR: 0.8 (Q1)

N. Arandia, J. I. Garate, and J. Mabe, "Embedded Sensor Systems in Medical Devices: Requisites and Challenges Ahead," Sensors, vol. 22, no. 24, p. 9917, Dec. 2022, doi: 10.3390/s22249917.

Abstract: The evolution of technology enables the design of smarter medical devices. Embedded Sensor Systems play an important role, both in monitoring and diagnostic devices for healthcare. The design and development of Embedded Sensor Systems for medical devices are subjected to standards and regulations that will depend on the intended use of the device as well as the used technology. This article summarizes the challenges to be faced when designing Embedded Sensor Systems for the medical sector. With this aim, it presents the innovation context of the sector, the stages of new medical device development, the technological components that make up an Embedded Sensor System and the regulatory framework that applies to it. Finally, this article highlights the need to define new medical product design and development methodologies that help companies to successfully introduce new technologies in medical devices.

7.3.2 Monitoring of vital signs in the home environment: A review of current technologies and solutions

Conference Paper
Date of publication: 16 February 2023
The 16th International Joint Conference on Biomedical Engineering Systems and Technologies

N. Arandia, N, J.I. Garate, J. Mabe, "Monitoring of Vital Signs in the Home Environment: A Review of Current Technologies and Solutions", In Proceedings of the 16th International Joint Conference on Biomedical Engineering Systems and Technologies - BIODEVICES, ISBN 978-989-758-631-6; ISSN 2184-4305, pages 108-115. DOI: 10.5220/0011646700003414.

Abstract: Vital signs measurement is key for monitoring and controlling the health of patients in the home environment. Parameters such as body temperature, heart rate, blood pressure, respiratory rate, oxygen saturation or blood glucose reflect the state of essential functions of the human body. Deviations of some of these parameters may indicate illness or worsening of the patient's condition. Nowadays there are different devices that allow the measurement of the main vital signs, in this article the measurement technologies as well as the main medical devices are reviewed. Many of these devices are not suitable for simultaneous monitoring of several vital signs so the patient is required to handle a multitude of devices. Therefore, a review of new monitoring device concepts that combine more than one vital sign and do not interfere with the day-to-day life of patients is carried out.

7.3.3 Medical Devices with Embedded Sensor Systems: Design and Development Methodology for Start-Ups

Journal Paper
Date of publication: 26 February 2023
Sensors MDPI (ISSN 1424-8220), Section Biomedical Sensors
Special Issue Embedded Sensor Systems for Health
Impact Factor 2021: 3.847
CiteScore 2021: 6.4
SJR: 0.8 (Q1)

N. Arandia, J. I. Garate, and J. Mabe, "Medical Devices with Embedded Sensor Systems: Design and Development Methodology for Start-Ups," Sensors, vol. 23, no. 5, p. 2578, Feb. 2023, doi: 10.3390/s23052578.

Abstract: Embedded systems have become a key technology for the evolution of medical devices. However, the regulatory requirements that must be met make designing and developing these devices challenging. As a result, many start-ups attempting to develop medical devices fail. Therefore, this article presents a methodology to design and develop embedded medical devices while minimising the economic investment during the technical risk stages and encouraging customer feedback. The proposed methodology is based on the execution of three stages: Development Feasibility, Incremental and Iterative Prototyping, and Medical Product Consolidation. All this is completed in compliance with the applicable regulations. The methodology mentioned above is validated through practical use cases in which the development of a wearable device for monitoring vital signs is the most relevant. The presented use cases sustain the proposed methodology, for the devices were successfully CE marked. Moreover, ISO 13485 certification is obtained by following the proposed procedures.

7.3.4 Cutting-edge innovation on Artificial Intelligence for integrated care on neuromusculoskeletal disorders

Poster Abstract
Date of publication: 4 November 2022
22nd International Conference on Integrated Care, Odense, Denmark, 23-25 May 2022
International Journal of Integrated Care

Pedrosa, I., Quirós, P., de Rosario-Martínez, H., Garrido-Jaén, J.D., González, M.L., Sedano, J., Arandia, N. and Bazan, X., 2022. Cutting-edge innovation on Artificial Intelligence for integrated care on neuromusculoskeletal disorders. International Journal of Integrated Care, 22(S3), p.403. DOI: 10.5334/ijic.ICIC22370.

Abstract: Neuromusculoskeletal diseases represent a key demand on the healthcare system and society, due to the fact that their symptoms require continuous care and monitoring, especially

considering the ongoing demographic ageing trend. Nowadays, around 2% of the population in Spain suffers from one of these diseases. However, some of them are estimated to grow fourfold in the next 50 years. Added to this trend, is a relevant problem on disease-masking effects linked to the natural ageing process on these diseases and their consequences. Fortunately, the emergence and spreading of innovative technologies and infrastructures in the healthcare context is making it possible to promote the evolution of the health care model from intensive care to prevention. This new approach aims to empower the population as a key agent in managing, monitoring and guiding the processes that promote self-care, as the backbone of the real and effective development and deployment of so-called personalised medicine. Cutting-edge technologies offer the possibility of gathering continuous objective patient information, including Real World Data, being able to identify, at an early stage, any sign/symptom that may generate a risk or accelerate decline, and carry out an early personalised intervention to improve or extend quality of life. To this end, IBERUS is tackling the use of biomechanical assessment and smart health technologies that enable data gathering to apply Artificial Intelligence underpinned on a Big Data scalable architecture to:

- Provide new evidence on innovative indicators for clinical evaluation,
- Develop cutting-edge technologies to provide a functional characterisation and monitoring of people both in clinical settings and in their daily-life scenarios, as well as making possible to generate real-time responses of the assessment algorithms in rehabilitation,
- Define new criteria and methods for clinical decision support in treatment and rehabilitation,
- Applying third-generation Artificial Intelligence techniques integrating synthetic data and images of neuromusculoskeletal pathologies,
- Effectively share and manage Smart Health Data for the prognosis, treatment and care of neuromusculoskeletal pathologies, through innovative technologies based on the recording, integration, interoperability and advance data exploitation,

- Physicians and expert researchers will have an active involvement both at co-design and co-validation phases of several innovative high-impact solutions at a clinical and professional level.

As a result, emerging digital tools, innovative technical architectures and expert knowledge will be provided to improve clinical care and integrated care through new products and services for the diagnosis, rehabilitation, treatment and care of degenerative diseases of the neuromusculoskeletal system in clinical and out-of-hospital settings.

7.4 Future Works

Three main research lines are identified to continue the work carried out during this thesis:

- Manufacture of medical products:

Medical device manufacturers not only have to deal with the design and development phase but also with the manufacturing phase before placing their products on the market. This phase is critical as it is the last step to guarantee the quality and safety of the products. Therefore, completing the methodology with the manufacturing phase is considered interesting.

- Artificial Intelligence (AI) based medical devices:

Artificial Intelligence is increasingly present in healthcare solutions. In particular, it is used to support healthcare professionals in diagnostic decision-making. The peculiarities of this technology, such as decision-making based on automated learning, make it necessary to investigate new methodologies that allow it to be safely integrated into medical devices.

In February 2023, the International Electrotechnical Commission published the IEC 23894 guide to help organisations that develop, implement or use AI technology to manage risks. Therefore, introducing technical and regulatory requirements associated with using AI in medical devices is identified as a possible line of work for the future.

- Migrating proposed methodology to automotive embedded system development:

Developing embedded medical devices requires ensuring the quality and safety of the devices while complying with the sector's regulatory requirements. Similarly, other strictly regulated sectors, such as the automotive industry, can benefit from such methodologies.

The automotive sector is currently undergoing constant growth and change as the decarbonisation of transport will bring new, much smarter and connected vehicles. Embedded electronics can become a crucial technology in this process of endowing components with intelligence, thus converting purely mechanical components into electromechanical components.

This technological change implies that many companies that do not usually work with electronic circuits must incorporate this technology into their products. Therefore, it is expected that a methodology such as the one proposed in this thesis can facilitate this transition.

After an initial analysis, equivalences have been identified between the standards regulating the automotive and healthcare sectors. ISO 26262, the standard that ensures functional safety in the automotive sector, as well as IEC 62304 and IEC 60601-1 in the health sector, are derived from the IEC 61508 standard. Therefore, they share the core of design and development processes. For all these, this future research line is identified as a priority.

8. References

[1] 'Health product policy and standards'. https://www.who.int/teams/health-product-policy-and-standards/assistive-and-medical-technology/medical-devices (accessed Feb. 03, 2022).

[2] 'What is HealthTech? | LinkedIn'. https://www.linkedin.com/pulse/what-healthtech-jorge-juan-fernández-garcía/ (accessed Feb. 03, 2022).

[3] European Council and Parliament, 'REGULATION (EU) 2017/746 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on in vitro diagnostic medical devices', *Official Journal of the European Union*, 2017.

[4] U.S. Food and Drug Administration, 'Title 21 - Food and Drugs', *CFR - Code of Federal Regulations*, 2022.

[5] C. Robertson and V. Laurion, 'Tip of the Iceberg II: How the Intended-Uses Principle Produces Medical Knowledge and Protects Liberty', *New York University Journal of Law and Liberty*, vol. 11, 2017, Accessed: Nov. 19, 2022. [Online]. Available: https://heinonline.org/HOL/Page?handle=hein.journals/nyujlawlb11&id=780&div=&collection=

[6] S. Gehr and C. Russmann, 'Shaping the future of cardiovascular medicine in the new era of wearable devices',

Nature Reviews Cardiology 2022 19:8, vol. 19, no. 8, pp. 501–502, Jun. 2022, doi: 10.1038/s41569-022-00729-2.

[7] F. V Lima *et al.*, 'At the Crossroads! Time to Start Taking Smartwatches Seriously', *Am J Cardiol*, vol. 179, pp. 96–101, 2022, doi: https://doi.org/10.1016/j.amjcard.2022.06.020.

[8] C. (Chinh T. D. Bui, 'A study of the challenges faced by FDA-regulated early-stage medical device startups and how to approach them', 2020, Accessed: Nov. 19, 2022. [Online]. Available: https://dspace.mit.edu/handle/1721.1/132807

[9] D. B. Jefferys, 'The regulation of medical devices and the role of the Medical Devices Agency.', *Br J Clin Pharmacol*, vol. 52, no. 3, pp. 229–235, 2001, doi: 10.1046/j.0306-5251.2001.01416.x.

[10] C. Heneghan, M. Thompson, M. Billingsley, and D. Cohen, 'Medical-device recalls in the UK and the device-regulation process: retrospective review of safety notices and alerts', *BMJ Open*, vol. 1, no. 1, 2011, doi: 10.1136/BMJOPEN-2011-000155.

[11] A. D. McCarthy and P. V Lawford, 'Standalone medical device software: The evolving regulatory framework', *J Med Eng Technol*, vol. 39, no. 7, pp. 441–447, 2015, doi: 10.3109/03091902.2015.1088087.

[12] N. Carroll and I. Richardson, 'Software-as-a-Medical Device: demystifying Connected Health regulations', *Journal of Systems and Information Technology*, vol. 18, no. 2, pp. 186–215, 2016, doi: 10.1108/JSIT-07-2015-0061/FULL/XML.

[13] D. R. Petretto and R. Pili, 'Ageing and COVID-19: What Is the Role for Elderly People?', *Geriatrics*, vol. 5, no. 2, 2020, doi: 10.3390/geriatrics5020025.

[14] F. A. Rabi, M. S. Al Zoubi, G. A. Kasasbeh, D. M. Salameh, and A. D. Al-Nasser, 'SARS-CoV-2 and Coronavirus Disease 2019: What We Know So Far', *Pathogens*, vol. 9, no. 3, 2020, doi: 10.3390/pathogens9030231.

[15] S. Bahl, R. P. Singh, M. Javaid, I. H. Khan, R. Vaishya, and R. Suman, 'Telemedicine Technologies for Confronting COVID-19 Pandemic: A Review', *https://doi.org/10.1142/S2424862220300057*, vol. 5, no. 4, pp. 547–561, Oct. 2020, doi: 10.1142/S2424862220300057.

[16] K. Ruan, Z. Wu, and Q. Xu, 'Smart Cleaner: A New Autonomous Indoor Disinfection Robot for Combating the COVID-19 Pandemic', *Robotics*, vol. 10, no. 3, 2021, doi: 10.3390/robotics10030087.

[17] J. Foster, A. B. Lloyd, and G. Havenith, 'Non-contact infrared assessment of human body temperature: The journal Temperature toolbox', *Temperature*, vol. 8, no. 4, pp. 306–319, 2021, doi: 10.1080/23328940.2021.1899546.

[18] A. M. Luks and E. R. Swenson, 'Pulse oximetry for monitoring patients with COVID-19 at home potential pitfalls and practical guidance', *Ann Am Thorac Soc*, vol. 17, no. 9, pp. 1040–1046, Sep. 2020, doi: 10.1513/ANNALSATS.202005-418FR/SUPPL_FILE/DISCLOSURES.PDF.

[19] 'Ageing and health'. https://www.who.int/news-room/fact-sheets/detail/ageing-and-health (accessed Feb. 03, 2022).

[20] A. Alanazi, 'IoT and AI in healthcare: A Systematic Literature Review', *PLOMS AI*, vol. 1, no. 1, p. 1, May 2021, Accessed: Nov. 19, 2022. [Online]. Available: https://plomscience.com/journals/index.php/PLOMSAI/article/view/1

[21] D. Jennings, M. M. Little, and M. Johnson-Jennings, 'Developing a Tribal Health Sovereignty Model for Obesity Prevention', *Prog Community Health Partnersh*, vol. 12, no. 3, pp. 353–362, Sep. 2018, doi: 10.1353/CPR.2018.0059.

[22] S. U. D. Wani *et al.*, 'Utilization of Artificial Intelligence in Disease Prevention: Diagnosis, Treatment, and Implications for the Healthcare Workforce', *Healthcare*, vol. 10, no. 4, 2022, doi: 10.3390/healthcare10040608.

[23] R. F. Mansour, A. El Amraoui, I. Nouaouri, V. G. Díaz, D. Gupta, and S. Kumar, 'Artificial Intelligence and Internet of Things Enabled Disease Diagnosis Model for Smart Healthcare Systems', *IEEE Access*, vol. 9, pp. 45137–45146, 2021, doi: 10.1109/ACCESS.2021.3066365.

[24] M. H. Al Banna, T. Ghosh, K. A. Taher, M. S. Kaiser, and M. Mahmud, 'A Monitoring System for Patients of Autism Spectrum Disorder Using Artificial Intelligence', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12241 LNAI, 2020, doi: 10.1007/978-3-030-59277-6_23/FIGURES/5.

[25] X. Li *et al.*, 'Artificial intelligence-powered remote monitoring of patients with chronic obstructive pulmonary disease', *Chin Med J (Engl)*, vol. 134, no. 13, pp. 1546–1548, 2021, doi: 10.1097/CM9.0000000000001529.

[26] J. Troisi, 'Artificial Intelligence in Medicine: Disease Diagnosis, Drug Development and Treatment Personalization', *Curr Med Chem*, vol. 28, no. 32, pp. 6509–6511, Oct. 2021, doi: 10.2174/0929867328322109141027 19.

[27] M. Tahri Sqalli and D. Al-Thani, 'On How Chronic Conditions Affect the Patient-AI Interaction: A Literature Review', *Healthcare*, vol. 8, no. 3, 2020, doi: 10.3390/healthcare8030313.

[28] J. C. L. Chow, 'Artificial Intelligence in Radiotherapy and Patient Care', in *Artificial Intelligence in Medicine*, Cham: Springer International Publishing, 2022, pp. 1275–1286. doi: 10.1007/978-3-030-64573-1_143.

[29] H. Fouad, A. S. Hassanein, A. M. Soliman, and H. Al-Feel, 'Analyzing patient health information based on IoT sensor with AI for improving patient assistance in the future direction', *Measurement*, vol. 159, p. 107757, 2020, doi: https://doi.org/10.1016/j.measurement.2020.107757.

[30] K. Takei, W. Honda, S. Harada, T. Arie, and S. Akita, 'Toward Flexible and Wearable Human-Interactive Health-

Monitoring Devices', *Adv Healthc Mater*, vol. 4, no. 4, pp. 487–500, Mar. 2015, doi: 10.1002/ADHM.201400546.

[31] P.-T. Chen, C.-L. Lin, and W.-N. Wu, 'Big data management in healthcare: Adoption challenges and implications', *Int J Inf Manage*, vol. 53, p. 102078, 2020, doi: https://doi.org/10.1016/j.ijinfomgt.2020.102078.

[32] M. Chen, J. Yang, J. Zhou, Y. Hao, J. Zhang, and C.-H. Youn, '5G-Smart Diabetes: Toward Personalized Diabetes Diagnosis with Healthcare Big Data Clouds', *IEEE Communications Magazine*, vol. 56, no. 4, pp. 16–23, 2018, doi: 10.1109/MCOM.2018.1700788.

[33] S. Liu, D. Zhai, and B. Han, 'FPGA medical big data system and ischemic stroke rehabilitation nursing', *Microprocess Microsyst*, vol. 83, p. 104014, 2021, doi: https://doi.org/10.1016/j.micpro.2021.104014.

[34] E. Mezghani, E. Exposito, K. Drira, M. Da Silveira, and C. Pruski, 'A Semantic Big Data Platform for Integrating Heterogeneous Wearable Data in Healthcare', *J Med Syst*, vol. 39, no. 12, pp. 1–8, Dec. 2015, doi: 10.1007/S10916-015-0344-X/FIGURES/4.

[35] W.-Y. Hsu, Y.-W. Cheng, and C.-B. Tsai, 'An Effective Algorithm to Analyze the Optokinetic Nystagmus Waveforms from a Low-Cost Eye Tracker', *Healthcare*, vol. 10, no. 7, 2022, doi: 10.3390/healthcare10071281.

[36] O. Alvarado-Cando, G. Belén Jara, P. Barzallo, and H. Torres-Salamea, 'A software based on eye gaze to evaluate mathematics in children with cerebral palsy in inclusive education', *Advances in Intelligent Systems and Computing*, vol. 794, pp. 909–915, 2019, doi: 10.1007/978-3-319-94947-5_89/TABLES/1.

[37] Irisbond, 'Eye-tracking technology integration for business & companies | IRISBOND'. https://www.irisbond.com/en/eye-tracking-technology/ (accessed Feb. 03, 2022).

[38] E. C. Zehnder, G. M. Schmölzer, M. van Manen, and B. H. Y. Law, 'Using eye-tracking augmented cognitive task analysis to explore healthcare professionals' cognition during neonatal resuscitation', *Resusc Plus*, vol. 6, p. 100119, 2021, doi: https://doi.org/10.1016/j.resplu.2021.100119.

[39] B. H. Y. Law, P.-Y. Cheung, M. Wagner, S. van Os, B. Zheng, and G. Schmölzer, 'Analysis of neonatal resuscitation using eye tracking: a pilot study', *Arch Dis Child Fetal Neonatal Ed*, vol. 103, no. 1, pp. F82--F84, 2018, doi: 10.1136/archdischild-2017-313114.

[40] S. Wang *et al.*, 'Artificial Intelligence in Lung Cancer Pathology Image Analysis', *Cancers (Basel)*, vol. 11, no. 11, 2019, doi: 10.3390/cancers11111673.

[41] C. J. Mathew, A. M. David, and C. M. J. Mathew, 'Artificial Intelligence and its future potential in lung cancer screening', *EXCLI J*, vol. 19, p. 1552, 2020, doi: 10.17179/EXCLI2020-3095.

[42] A. A. Rabaan *et al.*, 'Artificial Intelligence for Clinical Diagnosis and Treatment of Prostate Cancer', *Cancers (Basel)*, vol. 14, no. 22, 2022, doi: 10.3390/cancers14225595.

[43] S. Minami *et al.*, 'Diagnosis of Depth of Submucosal Invasion in Colorectal Cancer with AI Using Deep Learning', *Cancers (Basel)*, vol. 14, no. 21, 2022, doi: 10.3390/cancers14215361.

[44] M. Madani, M. M. Behzadi, and S. Nabavi, 'The Role of Deep Learning in Advancing Breast Cancer Detection Using Different Imaging Modalities: A Systematic Review', *Cancers (Basel)*, vol. 14, no. 21, 2022, doi: 10.3390/cancers14215334.

[45] T. Szydło and M. Konieczny, 'Mobile and wearable devices in an open and universal system for remote patient monitoring', *Microprocess Microsyst*, vol. 46, pp. 44–54, 2016, doi: https://doi.org/10.1016/j.micpro.2016.07.006.

[46] M. Niswar, M. Nur, A. A. Ilham, and I. Mappangara, 'A low cost wearable medical device for vital signs monitoring in low-resource settings', *International Journal of Electrical and*

Computer Engineering (IJECE), vol. 9, no. 4, pp. 2321–2327, 2019, doi: 10.11591/ijece.v9i4.pp2321-2327.

[47] F. Wu, H. Zhao, Y. Zhao, and H. Zhong, 'Development of a Wearable-Sensor-Based fall detection system', *Int J Telemed Appl*, vol. 2015, Jan. 2015, doi: 10.1155/2015/576364.

[48] M. Saleh and R. L. B. Jeannès, 'Elderly Fall Detection Using Wearable Sensors: A Low Cost Highly Accurate Algorithm', *IEEE Sens J*, vol. 19, no. 8, pp. 3156–3164, 2019, doi: 10.1109/JSEN.2019.2891128.

[49] T. Tsuda, 'Epidemiology of fragility fractures and fall prevention in the elderly: a systematic review of the literature', *Curr Orthop Pract*, vol. 28, no. 6, p. 580, 2017, doi: 10.1097/BCO.0000000000000563.

[50] F. Hoffmann-La Roche Ltd, 'About Roche'. https://www.roche.com/about.htm (accessed Feb. 03, 2022).

[51] Medtronic, 'Key Facts of Medtronic'. https://www.medtronic.com/us-en/our-company/key-facts.html (accessed Feb. 03, 2022).

[52] Alcon, 'Third Quarter 2021 Results'. https://www.alcon.com/media-release/alcon-reports-third-quarter-2021-results (accessed Feb. 03, 2022).

[53] Siemens, 'Siemens Healthineers with strong finish in record year 2021'. https://www.siemens-healthineers.com/press/releases/2021q4 (accessed Feb. 03, 2022).

[54] Abbott, 'Quarter 2021 Results'. https://abbott.mediaroom.com/2022-01-26-Abbott-Reports-Strong-Fourth-Quarter-2021-Results-Issues-2022-Forecast (accessed Feb. 03, 2022).

[55] Medtronic, 'Hugo™ RAS System'. https://www.medtronic.com/covidien/en-gb/robotic-assisted-surgery/hugo-ras-system/products-and-system.html (accessed Feb. 03, 2022).

[56] Microsoft, 'Driving digital transformation for in vitro diagnostic medical devices with IoT'. https://customers.microsoft.com/it-it/story/roche-diagnostics (accessed Feb. 06, 2022).

[57] Microsoft, 'Roche and Microsoft Sign a Memorandum of Understanding to improve healthcare sector through technology'. https://news.microsoft.com/en-xm/2021/12/14/roche-and-microsoft-sign-a-memorandum-of-understanding-to-improve-healthcare-sector-through-technology/ (accessed Feb. 11, 2022).

[58] Asociación Española de Business Angels (AEBAN) and IESE Business School, 'La inversión en startups: actividad y tendencias'. 2020.

[59] J. V., '2021: un año dorado para las empresas de salud digital', *Planta Doce*, Dec. 27, 2021. https://www.plantadoce.com/empresa/2021-un-ano-dorado-para-las-empresas-de-salud-digital.html (accessed Nov. 19, 2022).

[60] Koa Health, 'About Koa Health'. https://www.koahealth.com/about-us/ (accessed Feb. 22, 2022).

[61] Inbrain Neuroelectronics, 'About Inbrain'. https://www.inbrain-neuroelectronics.com/ (accessed Feb. 03, 2022).

[62] MedLumics, 'About MedLumics'. https://www.medlumics.com/about-us/ (accessed Feb. 03, 2022).

[63] D. Dias and J. Paulo Silva Cunha, 'Wearable Health Devices—Vital Sign Monitoring, Systems and Technologies', *Sensors*, vol. 18, no. 8, 2018, doi: 10.3390/s18082414.

[64] N. Arandia, J. Garate, and J. Mabe, 'Monitoring of Vital Signs in the Home Environment: A Review of Current Technologies and Solutions', *Proceedings of the 16th International Joint Conference on Biomedical Engineering Systems and*

Technologies, pp. 108–115, 2023, doi: 10.5220/0011646700003414.

[65] S. Anand and V. Nath, 'Study and Design of Smart Embedded System for Remote Health Monitoring Using Internet of Things', *Lecture Notes in Electrical Engineering*, pp. 409–414, 2019.

[66] N. Hurvitz, H. Azmanov, A. Kesler, and Y. Ilan, 'Establishing a second-generation artificial intelligence-based system for improving diagnosis, treatment, and monitoring of patients with rare diseases', *European Journal of Human Genetics 2021 29:10*, vol. 29, no. 10, pp. 1485–1490, Jul. 2021, doi: 10.1038/s41431-021-00928-4.

[67] H. M. Abdul-Jabbar and J. K. Abed, 'Real Time Pacemaker Patient Monitoring System Based on Internet of Things', *IOP Conf Ser Mater Sci Eng*, vol. 745, no. 1, p. 12093, Mar. 2020, doi: 10.1088/1757-899X/745/1/012093.

[68] M. R. Neuman, 'Measurement of Vital Signs: Temperature [Tutorial]', *IEEE Pulse*, vol. 1, no. 2, pp. 40–49, 2010, doi: 10.1109/MPUL.2010.937907.

[69] World Health Organization, 'Hypertension'.

[70] J. Landgraf, S. H. Wishner, and R. A. Kloner, 'Comparison of Automated Oscillometric Versus Auscultatory Blood Pressure Measurement', *Am J Cardiol*, vol. 106, no. 3, pp. 386–388, 2010, doi: https://doi.org/10.1016/j.amjcard.2010.03.040.

[71] R. Lazazzera, Y. Belhaj, and G. Carrault, 'A New Wearable Device for Blood Pressure Estimation Using Photoplethysmogram', *Sensors*, vol. 19, no. 11, 2019, doi: 10.3390/s19112557.

[72] C. Gonzalez Viejo, S. Fuentes, D. D. Torrico, and F. R. Dunshea, 'Non-Contact Heart Rate and Blood Pressure Estimations from Video Analysis and Machine Learning Modelling Applied to Food Sensory Responses: A Case Study for Chocolate', *Sensors*, vol. 18, no. 6, 2018, doi: 10.3390/s18061802.

[73] X. Zhang, S. Long, Y. Song, and X. Chen, 'Design and Implementation of a Portable Electrocardiograph', in *2021 International Conference on Electronic Information Technology and Smart Agriculture (ICEITSA)*, 2021, pp. 23–26. doi: 10.1109/ICEITSA54226.2021.00014.

[74] O. S. Hoilett, A. M. Twibell, R. Srivastava, and J. C. Linnes, 'Kick LL: A Smartwatch for Monitoring Respiration and Heart Rate using Photoplethysmography', in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2018, pp. 3821–3824. doi: 10.1109/EMBC.2018.8513356.

[75] T. Inui *et al.*, 'Use of a Smart Watch for Early Detection of Paroxysmal Atrial Fibrillation: Validation Study', *JMIR Cardio 2020;4(1):e14857 https://cardio.jmir.org/2020/1/e14857*, vol. 4, no. 1, p. e14857, Jan. 2020, doi: 10.2196/14857.

[76] G. Lu, F. Yang, J. A. Taylor, and J. F. Stein, 'A comparison of photoplethysmography and ECG recording to analyse heart rate variability in healthy subjects', *J Med Eng Technol*, vol. 33, no. 8, pp. 634–641, 2009, doi: 10.3109/03091900903150998.

[77] X. Li, J. Chen, G. Zhao, and M. Pietikainen, 'Remote Heart Rate Measurement From Face Videos Under Realistic Situations'. pp. 4264–4271, 2014.

[78] Y. Nakayama, G. Sun, S. Abe, and T. Matsui, 'Non-contact measurement of respiratory and heart rates using a CMOS camera-equipped infrared camera for prompt infection screening at airport quarantine stations', in *2015 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, 2015, pp. 1–4. doi: 10.1109/CIVEMSA.2015.7158595.

[79] A. Sapra, A. Malik, and P. Bhandari, 'Vital Sign Assessment', *StatPearls*, May 2022.

[80] H. Yamagami Akihiko Tanaka *et al.*, 'Dovepress association between respiratory impedance measured by forced oscillation technique and exacerbations in patients with COPD', *International Journal of COPD*, pp. 13–79, 2018, doi: 10.2147/COPD.S146669.

[81] R. Abbasi-Kesbi, A. Valipour, and K. Imani, 'Cardiorespiratory system monitoring using a developed acoustic sensor', *Healthc Technol Lett*, vol. 5, no. 1, pp. 7–12, Feb. 2018, doi: 10.1049/HTL.2017.0012.

[82] O. H. G. Diniz and O. H. G. Diniz, 'Volumetric Capnography: History, Function and Clinical Uses', *Open Access Library Journal*, vol. 9, no. 8, pp. 1–7, Aug. 2022, doi: 10.4236/OALIB.1109175.

[83] H. Tanaka, M. Yokose, S. Takaki, T. Mihara, Y. Saigusa, and T. Goto, 'Evaluation of respiratory rate monitoring using a microwave Doppler sensor mounted on the ceiling of an intensive care unit: a prospective observational study', *J Clin Monit Comput*, vol. 36, no. 1, pp. 71–79, Feb. 2022, doi: 10.1007/S10877-021-00733-W/TABLES/4.

[84] E. L'Her, S. Nazir, V. Pateau, and D. Visvikis, 'Accuracy of noncontact surface imaging for tidal volume and respiratory rate measurements in the ICU', *J Clin Monit Comput*, vol. 36, no. 3, pp. 775–783, Jun. 2022, doi: 10.1007/S10877-021-00708-X/TABLES/3.

[85] L. Luo, X. She, J. Cao, Y. Zhang, Y. Li, and P. X. K. Song, 'Detection and Prediction of Ovulation From Body Temperature Measured by an In-Ear Wearable Thermometer', *IEEE Trans Biomed Eng*, vol. 67, no. 2, pp. 512–522, 2020, doi: 10.1109/TBME.2019.2916823.

[86] S. Li, 'Prediction of Body Temperature from Smart Pillow by Machine Learning', in *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2019, pp. 421–426. doi: 10.1109/ICMA.2019.8816226.

[87] J. Zhang, 'Development of a Non-contact Infrared Thermometer', in *Proceedings of the 2017 International*

Conference Advanced Engineering and Technology Research (AETR 2017), Atlantis Press, pp. 308–312. doi: https://doi.org/10.2991/aetr-17.2018.59.

[88] H. Heinze *et al.*, 'Application and Evaluation in the Healthcare Domain', in *Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 215–230. doi: 10.1007/978-3-642-34614-9_15.

[89] I. Chiuchisan and O. Geman, 'Trends in embedded systems for e-Health and biomedical applications', in *2016 International Conference and Exposition on Electrical and Power Engineering (EPE)*, 2016, pp. 304–308. doi: 10.1109/ICEPE.2016.7781352.

[90] T. Massey *et al.*, 'Towards Reconfigurable Embedded Medical Systems', in *2007 Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS-MDPnP 2007)*, 2007, pp. 178–180. doi: 10.1109/HCMDSS-MDPnP.2007.7.

[91] A. BALA, 'REVOLUTIONARY EMBEDDED SYSTEMS IN HEALTHCARE', *Bodhi International Journal of Research in Humanities, Arts and Science*, vol. 5, no. 1, pp. 62–66, 2021.

[92] S. Charleer *et al.*, 'Sustained Impact of Real-time Continuous Glucose Monitoring in Adults With Type 1 Diabetes on Insulin Pump Therapy: Results After the 24-Month RESCUE Study', *Diabetes Care*, vol. 43, no. 12, pp. 3016–3023, 2020, doi: 10.2337/dc20-1531.

[93] P. A. H. Williams and A. J. Woodward, 'Cybersecurity vulnerabilities in medical devices: a complex environment and multifaceted problem', *Med Devices (Auckl)*, vol. 8, p. 305, Jul. 2015, doi: 10.2147/MDER.S50048.

[94] S. Yuan, A. Fernando, and D. C. Klonoff, 'Standards for Medical Device Cybersecurity in 2018', *J Diabetes Sci Technol*, vol. 12, no. 4, pp. 743–746, 2018, doi: 10.1177/1932296818763634.

[95] S. M. Karunarathne, N. Saxena, and M. K. Khan, 'Security and Privacy in IoT Smart Healthcare', *IEEE Internet Comput*, vol. 25, no. 4, pp. 37–48, 2021, doi: 10.1109/MIC.2021.3051675.

[96] K. J. Stol, M. A. Babar, P. Avgeriou, and B. Fitzgerald, 'A comparative study of challenges in integrating Open Source Software and Inner Source Software', *Inf Softw Technol*, vol. 53, no. 12, pp. 1319–1336, Dec. 2011, doi: 10.1016/J.INFSOF.2011.06.007.

[97] J. U. Ocampo and P. C. Kaminski, 'Medical device development, from technical design to integrated product development', *J Med Eng Technol*, vol. 43, no. 5, pp. 287–304, 2019, doi: 10.1080/03091902.2019.1653393.

[98] K. Fearis and A. Petrie, 'Best practices in early phase medical device development: Engineering, prototyping, and the beginnings of a quality management system', *Surgery*, vol. 161, no. 3, pp. 571–575, 2017, doi: https://doi.org/10.1016/j.surg.2016.08.052.

[99] F. Geremia, 'Quality aspects for medical devices, quality system and certification process', *Microchemical Journal*, vol. 136, pp. 300–306, 2018, doi: https://doi.org/10.1016/j.microc.2017.04.018.

[100] B. Karthika and A. R. Vijayakumar, 'ISO 13485: Medical Devices -- Quality Management Systems, Requirements for Regulatory Purposes', in *Medical Device Guidelines and Regulations Handbook*, Cham: Springer International Publishing, 2022, pp. 19–29. doi: 10.1007/978-3-030-91855-2_2.

[101] C. Insight, 'The Top 12 Reasons Startups Fail', 2021.

[102] D. B. Audretsch and M. P. Feldman, 'Small-Firm Strategic Research Partnerships: The Case of Biotechnology', *Technol Anal Strateg Manag*, vol. 15, no. 2, 2003, doi: 10.1080/0953732032000051154.

[103] 'EPO - Healthcare innovation main driver of European patent applications in 2020'. https://www.epo.org/news-events/news/2021/20210316.html (accessed Apr. 06, 2023).

[104] S. M. Ben-Menahem, R. Nistor-Gallo, G. Macia, G. von Krogh, and J. Goldhahn, 'How the new European regulation on medical devices will affect innovation', *Nature Biomedical Engineering 2020 4:6*, vol. 4, no. 6, pp. 585–590, Mar. 2020, doi: 10.1038/s41551-020-0541-x.

[105] S. Kaule *et al.*, 'Medical Device Regulation and current challenges for the implementation of new technologies', *Current Directions in Biomedical Engineering*, vol. 6, no. 3, pp. 334–337, Sep. 2020, doi: 10.1515/CDBME-2020-3086/MACHINEREADABLECITATION/RIS.

[106] I. R. Bednár and I. N. Tarišková, 'INDICATORS OF STARTUP FAILURE', *International Scientific Journal 'Industry 4.0'*, vol. II, no. 5, pp. 238–240, 2017, Accessed: Apr. 06, 2023. [Online]. Available: http://www.eban.org/about-angel-investment/early-stage-investing-

[107] N. Arandia, J. I. Garate, and J. Mabe, 'Embedded Sensor Systems in Medical Devices: Requisites and Challenges Ahead', *Sensors 2022, Vol. 22, Page 9917*, vol. 22, no. 24, p. 9917, Dec. 2022, doi: 10.3390/S22249917.

[108] I. Pedrosa *et al.*, 'Cutting-edge innovation on Artificial Intelligence for integrated care on neuromusculoskeletal disorders', *Int J Integr Care*, vol. 22, no. S3, p. 403, Nov. 2022, doi: 10.5334/IJIC.ICIC22370.

[109] N. Arandia, J. I. Garate, and J. Mabe, 'Medical Devices with Embedded Sensor Systems: Design and Development Methodology for Start-Ups', *Sensors 2023, Vol. 23, Page 2578*, vol. 23, no. 5, p. 2578, Feb. 2023, doi: 10.3390/S23052578.

[110] R. Urquhart, 'Understanding The Performance Of Processor IP Cores', *Semiconductor Engineering*, Aug. 27, 2020. https://semiengineering.com/understanding-the-performance-of-processor-ip-cores/ (accessed May 27, 2022).

[111] Krishna. Kant, 'Microprocessors and microcontrollers : architecture, programming and system design 80851, 8086, 8051, 8096', p. 728, 2007.

[112] Y. Bai, 'Introduction to Microcontrollers and This Book', *Practical Microcontroller Engineering with ARM®Technology*, pp. 1–12, 2015, doi: 10.1002/9781119058397.CH01.

[113] J. Eyre and J. Bier, 'Evolution of DSP processors', *IEEE Signal Process Mag*, vol. 17, no. 2, pp. 43–50, 2000, doi: 10.1109/79.826411.

[114] W. Hwang, 'New trends in low power SoC design technologies', in *IEEE International [Systems-on-Chip] SOC Conference, 2003. Proceedings.*, 2003, pp. 422-. doi: 10.1109/SOC.2003.1241559.

[115] I. Kuon, R. Tessier, and J. Rose, 'FPGA architecture: Survey and challenges', *Foundations and Trends in Electronic Design Automation*, vol. 2, no. 2, pp. 135–253, 2007, doi: 10.1561/1000000005.

[116] T. Yaqoob, H. Abbas, and M. Atiquzzaman, 'Security Vulnerabilities, Attacks, Countermeasures, and Regulations of Networked Medical Devices—A Review', *IEEE Communications Surveys \& Tutorials*, vol. 21, no. 4, pp. 3723–3768, 2019, doi: 10.1109/COMST.2019.2914094.

[117] Variscite, 'A trusted provider of embedded platforms and solutions'. https://www.variscite.com/about/ (accessed Nov. 20, 2022).

[118] Ying Bai, 'Introduction to Microcontrollers and This Book', *Practical Microcontroller Engineering with ARM® Technology*, pp. 1–12, Dec. 2015, doi: 10.1002/9781119058397.CH01.

[119] Ying Bai, 'ARM® Microcontroller Architectures', *Practical Microcontroller Engineering with ARM® Technology*, pp. 13–82, Dec. 2015, doi: 10.1002/9781119058397.CH02.

[120] A. Ltd, 'Arm Cortex-M Processor Comparison Table', 2022. https://developer.arm.com/documentation/102787/0100 (accessed May 16, 2022).

[121] R. Pawson, 'The myth of the Harvard architecture', *IEEE Annals of the History of Computing*, pp. 1–1, 2022, doi: 10.1109/MAHC.2022.3175612.

[122] ARM Ltd., 'Arm Cortex-A Processor Comparison Table', 2022. https://developer.arm.com/documentation/102826/latest/ (accessed May 16, 2022).

[123] J. Eyre and J. Bier, 'Evolution of DSP processors', *IEEE Signal Process Mag*, vol. 17, no. 2, pp. 43–50, 2000, doi: 10.1109/79.826411.

[124] T. Instruments, 'Digital signal processors (DSPs) product selection | TI.com', *Portfolio*. https://www.ti.com/microcontrollers-mcus-processors/processors/digital-signal-processors/products.html (accessed May 22, 2022).

[125] A. Devices, 'Selection Table for Processsors and DSP | Parametric Search | Analog Devices', *Portfolio*. https://www.analog.com/en/parametricsearch/11128#/ (accessed May 22, 2022).

[126] I. Kuon, R. Tessier, and J. Rose, 'FPGA architecture: Survey and challenges', *Foundations and Trends in Electronic Design Automation*, vol. 2, no. 2, pp. 135–253, 2007, doi: 10.1561/1000000005.

[127] Intel, 'Intel® FPGA Product Catalog v21.3'.

[128] AMD - XILINX, 'FPGAs & 3D ICs'. https://www.xilinx.com/products/silicon-devices/fpga.html (accessed May 22, 2022).

[129] Wei Hwang, 'New trends in low power SoC design technologies', in *IEEE International [Systems-on-Chip] SOC Conference*, Institute of Electrical and Electronics Engineers (IEEE), Jan. 2004, p. 422. doi: 10.1109/SOC.2003.1241559.

[130] NXP, 'QorIQ® Processing Platforms: 64-bit Multicore SoCs | NXP Semiconductors'. https://www.nxp.com/products/processors-and-microcontrollers/power-architecture/qoriq-communication-processors:QORIQ_HOME (accessed May 28, 2022).

[131] Intel, 'Chipset and System-on-a-Chip (SoC) Reference for Intel® NUC'. https://www.intel.co.uk/content/www/uk/en/support/articles/000056236/intel-nuc.html (accessed May 28, 2022).

[132] AMD - XILINX, 'SoCs, MPSoCs and RFSoCs'. https://www.xilinx.com/products/silicon-devices/soc.html (accessed May 28, 2022).

[133] Qualcomm, 'All Products and Platforms | Chipsets and SoCs | Qualcomm'. https://www.qualcomm.com/products (accessed May 28, 2022).

[134] A. H. Payberah, *Embedded Operating Systems and Linux*. Springer, 2018.

[135] Embedded Staff, 'Comparing microcontroller real-time operating systems - Embedded.com', *Embedded*, Dec. 08, 2013. https://www.embedded.com/comparing-microcontroller-real-time-operating-systems/ (accessed May 28, 2022).

[136] Wind River, 'Medical Solutions'. https://www.windriver.com/solutions/medical (accessed Nov. 20, 2022).

[137] Green Hills, 'INTEGRITY Real-time Operating System'. https://www.ghs.com/products/rtos/integrity.html#max (accessed May 28, 2022).

[138] BlackBerry QNX, 'Medical Device Operating System | RTOS, Hypervisor Embedded Systems '. https://blackberry.qnx.com/en/industries/medical (accessed Nov. 20, 2022).

[139] Siemens, 'Nucleus SafetyCert', 2022.

[140] ‘SafeRTOS Features – often used with popular ARM Cortex M3, M4 processors’, *High Integrity Systems*, 2022. https://www.highintegritysystems.com/safertos/rtos-features/ (accessed May 28, 2022).

[141] A. Bombarda *et al.*, ‘Guidelines for the development of a critical software under emergency’, *Inf Softw Technol*, vol. 152, p. 107061, 2022, doi: https://doi.org/10.1016/j.infsof.2022.107061.

[142] Wind River, ‘Wind River Linux Product Overview | Wind River’. https://www.windriver.com/resource/wind-river-linux-product-overview (accessed May 28, 2022).

[143] Siemens, ‘Sokol Flex OS’, 2022.

[144] J. F. Scott, *Basic Properties of RAMs (Random Access Memories)*. Springer, Berlin, Heidelberg, 2000. doi: 10.1007/978-3-662-04307-3_2.

[145] M. Joodaki, *Memory Devices*. Springer, Berlin, Heidelberg, 2013. doi: 10.1007/978-3-642-31350-9_3.

[146] R. Dean Adams, *Non-Volatile Memories*. Boston: Springer, Boston, MA, 2003. doi: 10.1007/0-306-47972-9_7.

[147] C. Julien, A. Mauger, A. Vijh, and K. Zaghib, *Lithium Batteries: Science and Technology*. Springer International Publishing, 2015. doi: 10.1007/978-3-319-19108-9.

[148] H. Wenzl, ‘BATTERIES | Capacity’, in *Encyclopedia of Electrochemical Power Sources*, J. Garche, Ed., Amsterdam: Elsevier, 2009, pp. 395–400. doi: https://doi.org/10.1016/B978-044452745-5.00043-5.

[149] N. Abavi-Torghabeh, M. M. Kalantarian, S. Dehkhoda, and Z. Sadeghian, ‘A systematic evaluation of charge-discharge behaviors, performance, and rate-capability of Al-ion batteries’, *Electrochim Acta*, vol. 437, p. 141508, 2023, doi: https://doi.org/10.1016/j.electacta.2022.141508.

[150] T. Guena and P. Leblanc, ‘How Depth of Discharge Affects the Cycle Life of Lithium-Metal-Polymer Batteries’, in *INTELEC*

06 - Twenty-Eighth International Telecommunications Energy Conference, 2006, pp. 1–8. doi: 10.1109/INTLEC.2006.251641.

[151] W. Waag and D. U. Sauer, 'SECONDARY BATTERIES – LEAD– ACID SYSTEMS | State-of-Charge/Health', in *Encyclopedia of Electrochemical Power Sources*, J. Garche, Ed., Amsterdam: Elsevier, 2009, pp. 793–804. doi: https://doi.org/10.1016/B978-044452745-5.00149-0.

[152] M. I. Wahyuddin, P. S. Priambodo, and H. Sudibyo, 'Energy Inefficiency on Battery Internal Resistance and Applications', in *2018 4th International Conference on Nano Electronics Research and Education (ICNERE)*, 2018, pp. 1–4. doi: 10.1109/ICNERE.2018.8642600.

[153] A. S. Sener, 'Improving the Life-Cycle and SOC of the Battery of a Modular Electric Vehicle Using Ultra-Capacitor', in *2019 8th International Conference on Renewable Energy Research and Applications (ICRERA)*, 2019, pp. 611–614. doi: 10.1109/ICRERA47325.2019.8996616.

[154] S. Arora, A. T. Abkenar, S. G. Jayasinghe, and K. Tammi, 'Chapter 8 - Battery Management System: Charge Balancing and Temperature Control', in *Heavy-Duty Electric Vehicles*, S. Arora, A. T. Abkenar, S. G. Jayasinghe, and K. Tammi, Eds., Butterworth-Heinemann, 2021, pp. 173–203. doi: https://doi.org/10.1016/B978-0-12-818126-3.00005-1.

[155] Renesas, 'Battery Management System Tutorial | Renesas'. https://www.renesas.com/in/en/products/power-power-management/battery-management-system-tutorial (accessed May 28, 2022).

[156] Texas Instruments, 'Battery management ICs | TI.com'. https://www.ti.com/power-management/battery-management/overview.html (accessed May 28, 2022).

[157] Analog Devices, 'Battery Management '. https://www.analog.com/en/product-category/battery-management.html (accessed May 28, 2022).

[158] Richtek Technology, 'Power Management Components for Li-ion Battery Powered Applications'. https://www.richtek.com/battery-management/en/index.html (accessed May 28, 2022).

[159] Maxim Integrated, 'Battery Management Solutions'. https://www.maximintegrated.com/en/products/power/battery-management.html (accessed May 28, 2022).

[160] X. Wei and J. Liu, 'Power sources and electrical recharging strategies for implantable medical devices', *Frontiers of Energy and Power Engineering in China 2008 2:1*, vol. 2, no. 1, pp. 1–13, Mar. 2008, doi: 10.1007/S11708-008-0016-3.

[161] T. Kimpe and T. Tuytschaever, 'Increasing the Number of Gray Shades in Medical Display Systems—How Much is Enough?', *J Digit Imaging*, vol. 20, no. 4, p. 422, Dec. 2007, doi: 10.1007/S10278-006-1052-3.

[162] A. Badano, '13.2: Viewing Angle Comparison of IPS and VA Medical AMLCDs', *SID Symposium Digest of Technical Papers*, vol. 36, no. 1, pp. 192–195, May 2005, doi: 10.1889/1.2036398.

[163] Mélanie Gaillet, 'TFT-LCD Display', France, 2003.

[164] H.-K. Chung, H.-D. Kim, and B. Kristal, 'AMOLED Technology for Mobile Displays', *SID Symposium Digest of Technical Papers*, vol. 37, no. 1, pp. 1447–1450, Jun. 2006, doi: 10.1889/1.2433260.

[165] F. Batool, 'IJARCCE A Review Paper on: Organic Light-Emitting Diode (OLED) Technology and Applications', *International Journal of Advanced Research in Computer and Communication Engineering ISO*, vol. 3297, no. 11, 2007, doi: 10.17148/IJARCCE.2016.51132.

[166] N. Desmarais, 'E Ink and Digital Paper', 2003.

[167] R. Phares, 'Resistive Touch Screen Technology', *Handbook of Visual Display Technology*, pp. 1–20, 2015, doi: 10.1007/978-3-642-35947-7_203-2.

[168] T. Gray, *Projected Capacitive Touch Basics*. Springer International Publishing, 2019. doi: 10.1007/978-3-319-98392-9_2.

[169] L.-B. Chen, T.-H. Huang, A. R. See, J. Antonio, G. Choco, and K. Chandramohan, 'Touch, Texture and Haptic Feedback: A Review on How We Feel the World around Us', *Applied Sciences 2022, Vol. 12, Page 4686*, vol. 12, no. 9, p. 4686, May 2022, doi: 10.3390/APP12094686.

[170] J. W. Gooch, 'Piezo-Electric Effect', *Encyclopedic Dictionary of Polymers*, pp. 536–536, 2011, doi: 10.1007/978-1-4419-6247-8_8724.

[171] HL7 International, 'About Health Level Seven International'. https://www.hl7.org/about/index.cfm?ref=nav (accessed May 29, 2022).

[172] HL7 International, 'HL7 Standards - Section 1d: Version 2 (V2)'. https://www.hl7.org/implement/standards/product_section.cfm?section=13 (accessed May 29, 2022).

[173] HL7 International, 'HL7 Standards - Section 1e: Version 3 (V3)'. https://www.hl7.org/implement/standards/product_section.cfm?section=14 (accessed May 29, 2022).

[174] HL7 International, 'HL7 Standards - Section 1a: Clinical Document Architecture (CDA®) '. https://www.hl7.org/implement/standards/product_section.cfm?section=10 (accessed May 29, 2022).

[175] HL7 International, 'HL7 Standards - Section 1c: FHIR®'. https://www.hl7.org/implement/standards/product_section.cfm?section=12 (accessed May 29, 2022).

[176] HL7 International, 'HL7 Standards - Section 1g: CCOW'. https://www.hl7.org/implement/standards/product_section.cfm?section=16 (accessed May 29, 2022).

[177] M. Kasparick, S. Schlichting, F. Golatowski, and D. Timmermann, 'New IEEE 11073 standards for interoperable,

networked point-of-care Medical Devices', in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, Institute of Electrical and Electronics Engineers Inc., Nov. 2015, pp. 1721–1724. doi: 10.1109/EMBC.2015.7318709.

[178] International Standardization Organization, 'ISO 13606-1: Health informatics — Electronic health record communication. Reference Model'. p. 103, 2008.

[179] IEEE, '11073-10101-2019 - IEEE Standard for Health informatics--Point-of-care medical device communication - Part 10101 : Nomenclature.', 2019.

[180] IEEE, '11073-10201-2020 - ISO/IEC/IEEE International Standard - Health informatics--Device interoperability--Part 10201 Point-of-care medical device communication--Domain information model.', 2020.

[181] IEEE, '11073-20101-2004 - ISO/IEC/IEEE International Standard - Health informatics - Point-of-care medical device communication - Application profile - Base standard', 2004.

[182] IEEE, '11073-30200a-2011 Health informatics--Point-of-care medical device communication Part 30200: Transport profile--Cable connected Amendment 1.'.

[183] M. Clarke, J. De Folter, V. Verma, and H. Gokalp, 'Interoperable End-to-End Remote Patient Monitoring Platform Based on IEEE 11073 PHD and ZigBee Health Care Profile', *IEEE Trans Biomed Eng*, vol. 65, no. 5, pp. 1014–1025, May 2018, doi: 10.1109/TBME.2017.2732501.

[184] H. F. Badawi, F. Laamarti, and A. El Saddik, 'ISO/IEEE 11073 Personal Health Device (X73-PHD) Standards Compliant Systems: A Systematic Literature Review', *IEEE Access*, vol. 7, pp. 3062–3073, 2019, doi: 10.1109/ACCESS.2018.2886818.

[185] Bluetooth SIG, 'Blood Pressure Profile 1.1'. https://www.bluetooth.com/specifications/specs/blood-pressure-profile-1-1/ (accessed May 29, 2022).

[186] Bluetooth SIG, 'Continuous Glucose Monitoring Profile 1.0.1'. https://www.bluetooth.com/specifications/specs/continuous-glucose-monitoring-profile-1-0-1/ (accessed May 29, 2022).

[187] Bluetooth SIG, 'Glucose Profile 1.0'. https://www.bluetooth.com/specifications/specs/glucose-profile-1-0/ (accessed May 29, 2022).

[188] Bluetooth SIG, 'Heart Rate Profile 1.0'. https://www.bluetooth.com/specifications/specs/heart-rate-profile-1-0/ (accessed May 29, 2022).

[189] Bluetooth SIG, 'Health Thermometer Profile 1.0'. www.bluetooth.com/specifications/specs/heart-rate-profile-1-0/ (accessed May 29, 2022).

[190] Bluetooth SIG, 'Pulse Oximeter Profile 1.0.1'. www.bluetooth.com/specifications/specs/pulse-oximeter-profile-1-0 (accessed May 29, 2022).

[191] Bluetooth SIG, 'Weight Scale Profile 1.0'. www.bluetooth.com/specifications/specs/weight-scale-profile (accessed May 29, 2022).

[192] Bluetooth SIG, 'Insulin Delivery Profile 1.0.1'. https://www.bluetooth.com/specifications/specs/insulin-delivery-profile-1-0-1/ (accessed May 29, 2022).

[193] Clinical and Laboratory Standards Institute, 'POCT01-A2 Point-of-Care Connectivity'. 2006.

[194] ASTM International, 'About Us', 2022. https://www.astm.org/about/overview.html (accessed May 29, 2022).

[195] P. Mountain, D. Chou, J. Callaghan, R. Davis, C. D. Hawker, and D. Herold, *Specification for Transferring Information Between Clinical Laboratory Instruments and Information Systems*, vol. 24, no. 33. 2004.

[196] International Organization for Standardization, 'ISO 11073-90101:2008(en) Health informatics — Point-of-care medical

device communication — Part 90101: Analytical instruments — Point-of-care test'. 2008.

[197] IEEE, 'IEEE Std 802.3-2015 (Revision of IEEE Std 802.3-2012) : IEEE Standard for Ethernet.', 2016.

[198] IEEE, 'IEEE Standard for Information technology — Telecommunications and information Local and metropolitan area networks — Part 11 : Wireless LAN Medium Access Control (MAC)', *Control*, pp. C1-1184, 2007.

[199] IEEE Computer Society. LAN/MAN Standards Committee., Institute of Electrical and Electronics Engineers., IEEE-SA Standards Board., and American National Standards Institute., 'IEEE standard for information technology : telecommunications and information exchange between systems-- local and metropolitan area networks-- specific requirements. Part 15.1, Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications ', p. 580, 2005.

[200] M. Donati, A. Celli, A. Ruiu, S. Saponara, and L. Fanucci, 'A telemedicine service platform exploiting BT/BLE wearable sensors for remote monitoring of chronic patients', *2018 7th International Conference on Modern Circuits and Systems Technologies, MOCAST 2018*, pp. 1–4, Jun. 2018, doi: 10.1109/MOCAST.2018.8376643.

[201] A. Bhawiyuga, E. S. Pramukantoro, and A. P. Kirana, 'A web of thing middleware for enabling standard web access over BLE based healthcare wearable device', *2019 IEEE 1st Global Conference on Life Sciences and Technologies, LifeTech 2019*, pp. 265–267, Mar. 2019, doi: 10.1109/LIFETECH.2019.8883981.

[202] H. Kim *et al.*, 'Design of a Low-Power BLE5-Based Wearable Device for Tracking Movements of Football Players', *Proceedings - 2019 International SoC Design Conference, ISOCC 2019*, pp. 11–12, Oct. 2019, doi: 10.1109/ISOCC47750.2019.9027630.

[203] C. Gomez, J. Oller, and J. Paradells, 'Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-

Power Wireless Technology', *Sensors 2012, Vol. 12, Pages 11734-11753*, vol. 12, no. 9, pp. 11734–11753, Aug. 2012, doi: 10.3390/S120911734.

[204] A. I. Mondal and B. K. Mandal, 'Architecture of Bluetooth Security for Pairing Key and Better Authentication', *2021 5th International Conference on Information Systems and Computer Networks, ISCON 2021*, 2021, doi: 10.1109/ISCON52037.2021.9702335.

[205] Official Journal of the European Union, 'Council Directive 90/385/EEC of 20 June 1990 on the approximation of the laws of the Member States relating to active implantable medical devices'. pp. 1–20, 1990.

[206] Official Journal of the European Union, 'Council Directive 93/42/EEC of 14 June 1993 concerning medical devices'. pp. 1–60, 1993.

[207] Official Journal of the European Union, 'Directive 98/79/EC of European Parliament and of the Council of 27 October 1998 on in vitro diagnostic medical devices'. pp. 1–37, 1998.

[208] European Council and Parliament, 'REGULATION (EU) 2017/745 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on medical devices', *Official Journal of the European Union*, 2017.

[209] European Council and Parliament, 'REGULATION (EU) 2017/746 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on in vitro diagnostic medical devices', *Official Journal of the European Union*, 2017.

[210] International Electrotechnical Commission, 'IEC 60601-1 history and work in progress', *IEC Webstore*. https://webstore.iec.ch/publication/67497 (accessed Jul. 17, 2022).

[211] International Electrotechnical Commission, 'IEC TR 60601-4-5:2021 - Details and history', *IEC Webstore*, Jan. 18, 2021. https://webstore.iec.ch/publication/64703 (accessed Jul. 22, 2022).

[212] International Electrotechnical Commission, 'IEC 62304:2006+AMD1:2015 - Medical device software - Software life cycle processes'. 2015.

[213] International Organization for Standardization, 'ISO 14971:2019: Application of risk management to medical devices'. p. 36, 2019.

[214] International Electrotechnical Commission, 'IEC 62366-1:2015+AMD1:2020 - Medical devices - Part 1: Application of usability engineering to medical devices'. 2020.

[215] International Standarization Organization, 'ISO 13485:2016(en), Medical devices — Quality management systems — Requirements for regulatory purposes'. 2016.

[216] International Electrotechnical Commission, 'IEC 81001-5-1:2021 - Health software and health IT systems safety, effectiveness and security - Part 5-1: Security - Activities in the product life cycle'. Dec. .

[217] International Electrotechnical Commission, 'IEC 62133-2:2017+AMD1:2021 - Secondary cells and batteries containing alkaline or other non-acid electrolytes - Safety requirements for portable sealed secondary cells, and for batteries made from them, for use in portable applications - Part 2: Lithium s'. pp. 1–190, 2021.

[218] International Electrotechnical Commission, 'IEC 60086-4:2019 - Primary batteries - Part 4: Safety of lithium batteries '. pp. 1–90, 2019.

[219] T. Maak and J. Wylie, 'Medical Device Regulation: A Comparison of the United States and the European Union', *Journal of the American Academy of Orthopaedic Surgeons*, vol. 24, p. 1, 2016, doi: 10.5435/JAAOS-D-15-00403.

[220] European Council and Parliament, 'REGULATION (EU) 2017/745 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on medical devices', *Official Journal of the European Union*, May 2017.

[221] European Commission, 'Commission Implementing Decision of 14.4.2021 on a standardisation request to the European Committee for Standardization and the European Committee for Electrotechnical Standardization as regards medical devices in support of Regulation (EU) 2017/745 of t'. pp. 1–43, 2021.

[222] International Electrotechnical Commission, 'IEC 60601-1:2005+AMD1:2012+AMD2:2020 - Medical electrical equipment -- Part 1: General requirements for basic safety and essential performance'. Aug. 20, 2020.

[223] International Electrotechnical Commission, 'IEC 62304 history and work in progress', *IEC Webstore*. https://webstore.iec.ch/publication/22794 (accessed Jul. 18, 2022).

[224] UL Solutions, 'Things to Know About ISO 14971, 3rd Edition', Sep. 2020. https://www.ul.com/news/things-know-about-iso-14971-3rd-edition (accessed Jul. 19, 2022).

[225] International Electrotechnical Commission, 'IEC 62366-1: history and work in progress', *IEC Webstore*. https://webstore.iec.ch/publication/67220 (accessed Jul. 20, 2022).

[226] International Electrotechnical Commission, 'IEC TR 62366-2: history and work in progress', *IEC Webstore*, 2020. https://webstore.iec.ch/publication/24664 (accessed Jul. 20, 2022).

[227] International Electrotechnical Commission, 'IEC TR 62366-2:2016 - Medical devices - Part 2: Guidance on the application of usability engineering to medical devices'. 2016.

[228] S. Ramakrishna, L. Tian, C. Wang, S. Liao, and W. E. Teo, 'Medical Devices: Regulations, Standards and Practices', *Medical Devices: Regulations, Standards and Practices*, pp. 1–236, Oct. 2015, doi: 10.1016/C2014-0-02797-9.

[229] M. Torres and R. Bharwani, 'FDA Proposal to Align its Quality Systems with International Consensus Standard Will

Benefit Industry and Other Regulators', *FDA*, 2022. https://www.fda.gov/international-programs/international-programs-news-speeches-and-publications/fda-proposal-align-its-quality-systems-international-consensus-standard-will-benefit-industry-and (accessed Jun. 27, 2022).

[230] Council of the European Communities, 'Council Directive 93/42/EEC of 14 June 1993 concerning medical devices', *EUR-Lex*, Jun. 14, 1993. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:31993L0042&from=ES (accessed Jul. 22, 2022).

[231] International Electrotechnical Commission, 'IEC 80001-1:2021 - Details and History', *IEC Webstore*, Sep. 21, 2021. https://webstore.iec.ch/publication/34263 (accessed Jul. 22, 2022).

[232] U.S Department of Health and Human Services, 'Recognized Consensus Standards', *U.S Food and Drug Administration*, Jul. 06, 2018. https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfstandards/detail.cfm?standard__identification_no=37046 (accessed Jul. 22, 2022).

[233] U.S. Department of Health and Human Services; Food and Drug Administration; Center for Devices and Radiological Health; Center for Biologics Evaluation and Research, 'Implementation Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket Submissions Draft Guidance for Industry and Food and Drug Administration Staff', *U.S. Food and Drug Administration*, Apr. 2022.

[234] Johner Institute, 'IEC 81001-5-1: The standard for secure health software'. https://www.johner-institute.com/articles/software-iec-62304/and-more/iec-81001-5-1-the-standard-for-secure-health-software/ (accessed Jul. 22, 2022).

[235] P. Byrne, 'Application firewalls in a defence-in-depth design', *Network Security*, vol. 2006, no. 9, pp. 9–11, 2006, doi: https://doi.org/10.1016/S1353-4858(06)70422-6.

[236] Intertek, 'The Future of Battery Technologies-Part IV Testing & Evaluating Lithium Batteries', *The Future of Battery Technologies*.

[237] M. A. Akbar *et al.*, 'Statistical Analysis of the Effects of Heavyweight and Lightweight Methodologies on the Six-Pointed Star Model', *IEEE Access*, vol. 6, pp. 8066–8079, Feb. 2018, doi: 10.1109/ACCESS.2018.2805702.

[238] A. Chandrababu, 'A Comparison between Agile and Traditional Soware Development Methodologies'.

[239] R. Sherman, 'Project Management', *Business Intelligence Guidebook*, pp. 449–492, 2015, doi: 10.1016/B978-0-12-411461-6.00018-6.

[240] T. Thesing, C. Feldmann, and M. Burchardt, 'Agile versus Waterfall Project Management: Decision Model for Selecting the Appropriate Approach to a Project', *Procedia Comput Sci*, vol. 181, pp. 746–756, Jan. 2021, doi: 10.1016/J.PROCS.2021.01.227.

[241] J. Charvat, 'Project Management Methodologies: Selecting, Implementing, and Supporting Methodologies and Processes for Projects', *John Wiley & Sons*, p. 127, 2003.

[242] K. Chari and M. Agrawal, 'Impact of incorrect and new requirements on waterfall software project outcomes', *Empir Softw Eng*, vol. 23, no. 1, pp. 165–185, Feb. 2018, doi: 10.1007/S10664-017-9506-4/FIGURES/6.

[243] J. A. Osorio, M. R. V. Chaudron, and W. Heijstek, 'Moving from waterfall to iterative development - An empirical evaluation of advantages, disadvantages and risks of RUP', *Proceedings - 37th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2011*, pp. 453–460, 2011, doi: 10.1109/SEAA.2011.69.

[244] A. M. Dima and M. A. Maassen, 'From Waterfall to Agile software: Development models in the IT sector', *Journal of International Studies*, vol. 11, no. 2, pp. 315–326, 2018, doi: 10.14254/2071.

[245] L. Chen, P. Cooper, and Q. Liu, 'Security in Bluetooth Networks and Communications', *Wireless Network Security: Theories and Applications*, vol. 9783642365, pp. 77–94, Jun. 2013, doi: 10.1007/978-3-642-36511-9_5.

[246] T. Koivukangas, J. Lukkari, J. Haapola, and R. Kohno, 'Streamlining medical device design development process from research laboratory to the market', *International Symposium on Medical Information and Communication Technology, ISMICT*, vol. 2015-May, pp. 98–102, May 2015, doi: 10.1109/ISMICT.2015.7107506.

[247] S. Mathur Shaily Malik Asst Professor and C. Dept, 'Advancements in the V-Model', *©2010 International Journal of Computer Applications*, vol. 1, no. 12, pp. 975–8887.

[248] B. Liu, H. Zhang, and S. Zhu, 'An incremental V-model process for automotive development', *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, pp. 225–232, Jul. 2016, doi: 10.1109/APSEC.2016.040.

[249] A. Villafiorita, *Introduction to software project management*. Auerbach Publications, 2014.

[250] P. Jawandhiya *et al.*, 'Variations in V Model for Software Development', *International Journal of Advanced Research in Computer Science*, vol. 1, no. 2.

[251] R. Moraes, T. Basso, and E. Martins, 'V-Model Adaptation for Space Systems in Light of the ECSS Standard', *2021 10th Latin-American Symposium on Dependable Computing, LADC 2021 - Proceedings*, 2021, doi: 10.1109/LADC53747.2021.9672593.

[252] B. Liu, H. Zhang, and S. Zhu, 'An incremental V-model process for automotive development', *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, vol. 0, pp. 225–232, Jul. 2016, doi: 10.1109/APSEC.2016.040.

[253] K. Alexander and P. J. Clarkson, 'A validation model for the medical devices industry', *Journal of Engineering Design*, vol.

13, no. 3, pp. 197–204, Sep. 2002, doi: 10.1080/09544820110108890.

[254] I. Gräßler, D. Wiechel, D. Roesmann, and H. Thiele, 'V-model based development of cyber-physical systems and cyber-physical production systems', *Procedia CIRP*, vol. 100, pp. 253–258, Jan. 2021, doi: 10.1016/J.PROCIR.2021.05.119.

[255] I. Graessler and J. Hentze, 'The new V-Model of VDI 2206 and its validation das Neue V-Modell der VDI 2206 und seine Validierung', *At-Automatisierungstechnik*, vol. 68, no. 5, pp. 312–324, May 2020, doi: 10.1515/AUTO-2020-0015/MACHINEREADABLECITATION/RIS.

[256] Jürgen Gausemeier and Stefan Moehringer, 'New guideline Vdi 2206-a flexible procedure model for the design of mechatronic systems', in *International Conference on Engineering Design ICED 03*, Stockholm, Aug. 2003.

[257] Verein Deutscher Ingenieure (VDI), *VDI 2206: Design methodology for mechatronic systems*, 2004th–06 ed. 2004.

[258] P. S. Ganney, S. Pisharody, and E. Claridge, 'Software engineering', *Clin Eng*, pp. 131–168, 2020, doi: 10.1016/B978-0-08-102694-6.00009-7.

[259] A. Alshamrani, R. Qureshi, and A. Bahattab, 'A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model Related papers A Comprehensive St udy of Commonly Pract iced Heavy and Light Weight Soft ware Met hodolo… A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model', *IJCSI International Journal of Computer Science Issues*, vol. 12, no. 1, 2015.

[260] A. Mishra and D. Dubey, 'A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios Theoretical Analysis of Genetic Algorithms View project Robotics View project A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios', *International Journal of Advance*

Research in Computer Science and Management Studies, vol. 1, no. 5, 2013.

[261] A. V. Patil, 'Comparative Analysis between Sequential and Iterative Project Management Approaches', *International Research Journal of Engineering and Technology*, 2019.

[262] Connie M. Borror, 'The Certified Quality Engineer Handbook - Third Edition', pp. 167–171, 2008.

[263] A. Rathi and M. V. Krishna, 'A Pipelined Approach for Iterative Software Process Model', in *9th International Radar Symposium India* , India, 2013.

[264] A. Nguven-Duc, Y. Dahle, M. Steinert, and P. Abrahamsson, 'Towards understanding startup product development as effectual entrepreneurial behaviors', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10611 LNCS, pp. 265–279, 2017, doi: 10.1007/978-3-319-69926-4_19/FIGURES/3.

[265] A. Nilsson and T. L. Wilson, 'Reflections on Barry W. Boehm's "A spiral model of software development and enhancement"', *International Journal of Managing Projects in Business*, vol. 5, no. 4, pp. 737–756, Sep. 2012, doi: 10.1108/17538371211269031/FULL/XML.

[266] B. W. Boehm, 'A Spiral Model of Software Development and Enhancement', *Computer (Long Beach Calif)*, vol. 21, no. 5, pp. 61–72, 1988, doi: 10.1109/2.59.

[267] D. Doshi and L. Jain, 'REVIEW OF THE SPIRAL MODEL AND ITS APPLICATIONS', *International Journal of Engineering Applied Sciences and Technology*, vol. 5, pp. 311–316, 2021.

[268] R. Budde, K. Kautz, K. Kuhlenkamp, and H. Züllighoven, 'What is Prototyping?', *Prototyping*, pp. 6–9, 1992, doi: 10.1007/978-3-642-76820-0_2.

[269] I. H. Sarker, F. Faruque, U. Hossen, and A. Rahman, 'A survey of software development process models in software

engineering', *International Journal of Software Engineering and its Applications*, vol. 9, no. 11, pp. 55–70, 2015, doi: 10.14257/IJSEIA.2015.9.11.05.

[270] F. Kordon and Luqi, 'An introduction to rapid system prototyping', *IEEE Transactions on Software Engineering*, vol. 28, no. 9, pp. 817–821, Sep. 2002, doi: 10.1109/TSE.2002.1033222.

[271] S. Hekmatpour, 'Experience with evolutionary prototyping in a large software project', *ACM SIGSOFT Software Engineering Notes*, vol. 12, no. 1, pp. 38–41, Jan. 1987, doi: 10.1145/24574.24577.

[272] P. Pulli and R. Elmstrøm, 'IPTES — Incremental Prototyping Technology for Embedded real-time Systems', *Microprocessing and Microprogramming*, vol. 35, no. 1–5, pp. 13–21, Sep. 1992, doi: 10.1016/0165-6074(92)90288-I.

[273] J. A. Crowder and S. Friess, 'Introduction: The Agile Manager', *Agile Project Management: Managing for Success*, pp. 1–8, 2015, doi: 10.1007/978-3-319-09018-4_1.

[274] M. Todorović, D. Toljaga-Nikolić, and D. Bjelica, 'People-Oriented Principles and Values of Agile Project Management', *European Project Management Journal*, vol. 8, no. 2, pp. 3–8, 2018, doi: 10.18485/EPMJ.2018.8.2.1.

[275] Agile Alliance, 'About Agile Alliance', *Agile Alliance Website*. https://www.agilealliance.org/the-alliance/ (accessed Aug. 17, 2022).

[276] Agile Alliance, 'Agile Manifesto for Software Development'. https://www.agilealliance.org/agile101/the-agile-manifesto/ (accessed Aug. 09, 2022).

[277] Agile Alliance, 'Agile Glossary and Terminology', *Agile Alliance Website*. https://www.agilealliance.org/agile101/agile-glossary/ (accessed Aug. 17, 2022).

[278] Hirotaka Takeuchi and Ikujiro Nonaka, 'New New Product Development Game', *Harv Bus Rev*, Jan. 1986.

[279] J. Sutherland, 'Business object design and implementation workshop', *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA*, vol. 15-19-October-1995, pp. 170–175, Oct. 1995, doi: 10.1145/260094.260274.

[280] Rituraj Patil, *Mastering Advances Scrum Advanced Scrum Techniques for Scrum Teams, Roles, Artifacts, Events, Metrics, Working Agreements, Advanced Engineering Practices, and Technical Agility)*. BPB Publications, 2021.

[281] K. Schwaber and J. Sutherland, 'Scrum Team', *The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game*, p. 5.7, 2020.

[282] A. Dimitrescu, C. Babiş, E. Niculae, O. Chivu, and L. Dascălu, 'Efficiency of a production line by application of the Kanban method', *Journal of Research and Innovation for Sustainable Society (JRISS)*, vol. 1, no. 1, 2019, doi: 10.33727/JRISS.2019.1.4:29-34.

[283] Ľ. Dulina, M. Rakyta, I. Sulírová, and M. Šeligová, 'Improvement of the Production System Based on the Kanban Principle', *EAI/Springer Innovations in Communication and Computing*, pp. 199–216, 2019, doi: 10.1007/978-3-319-76998-1_14/COVER.

[284] D. J. Anderson, *Kanban: successful evolutionary change in your technology business*. Blue Hole Press, 2007.

[285] S. Bhuvana, K. Andrews, and D. Hemanand, 'Agile Software Methodology with Tools and Improvements', *Journal of Microcontroller Engineering and Applications*, vol. 8, no. 3, 2021, doi: 10.37591/JoMEA.

[286] A. Cockburn, *What is the grounding for Crystal?* Addison-Wesley Professional, 2004.

[287] A. Cockburn, 'What is the Crystal Family?', in *Crystal Clear: A Human-Powered Methodology for Small Teams: A Human-Powered Methodology for Small Teams*, Addison-Wesley Professional, 2004, pp. 239–241.

[288] A. Cockburn, 'Applied (The Seven Properties)', in *Crystal Clear: A Human-Powered Methodology for Small Teams: A Human-Powered Methodology for Small Teams*, Addison-Wesley Professional, 2004, pp. 33–62.

[289] A. Cockburn, 'Examined (The Work Products)', in *Crystal Clear: A Human-Powered Methodology for Small Teams: A Human-Powered Methodology for Small Teams*, Addison-Wesley Professional, 2004, pp. 151–216.

[290] A. Cockburn, 'Explored (The Process)', in *Crystal Clear: A Human-Powered Methodology for Small Teams: A Human-Powered Methodology for Small Teams2*, Addison-Wesley Professional, 2004, pp. 129–150.

[291] P. Y. Dave, 'The History of Lean Manufacturing by the view of Toyota-Ford', *Int J Sci Eng Res*, vol. 11, no. 8, pp. 1598–1602, Aug. 2020.

[292] J. P. Womack, D. T. Jones, D. Roos, and F. Sanders, *The machine that changed the world : the story of Lean Production, Toyota's secret weapon in the global car wars that is revolutionizing world industry*. Free Press, 1990.

[293] E. Ries, *The lean startup how today's entrepreneurs use continuous innovation to create radically successful businesses*, 1st ed. Currency, 2011.

[294] V. Lenarduzzi and D. Taibi, 'MVP Explained: A Systematic Mapping Study on the Definitions of Minimal Viable Product', *Proceedings - 42nd Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2016*, pp. 112–119, Oct. 2016, doi: 10.1109/SEAA.2016.56.

[295] A. Miski, 'Development of a Mobile Application Using the Lean Startup Methodology', *Int J Sci Eng Res*, vol. 5, no. 1, 2014.

[296] H. Edison, 'Lean internal startups: Challenges and lessons learned', *Fundamentals of Software Startups: Essential Engineering and Business Aspects*, pp. 251–268, Feb. 2020, doi: 10.1007/978-3-030-35983-6_15/COVER.

[297] T. Brown, 'Design Thinking', *Harv Bus Rev*, pp. 1–9, 2008.

[298] L. A. Deitte and R. A. Omary, 'The Power of Design Thinking in Medical Education', *Acad Radiol*, vol. 26, no. 10, pp. 1417–1420, Oct. 2019, doi: 10.1016/J.ACRA.2019.02.012.

[299] FJORD (Accenture), 'Design Thinking at FJORD'. https://www.accenture.com/t20180116t105743z__w__/fi-en/_acnmedia/pdf-69/accenture-fjord-servicedesign.pdf (accessed Aug. 13, 2022).

[300] SAP, 'A Combined Methodology to Build Scalable Solutions', *SAP - AppHaus Webpage*. https://apphaus.sap.com/approach (accessed Aug. 13, 2022).

[301] IDEO, 'IDEO Design Thinking | IDEO | Design Thinking', *IDEO Webpage*. https://designthinking.ideo.com/ (accessed Aug. 13, 2022).

[302] Google Ventures, 'The Design Sprint', *GV Webpage*. https://www.gv.com/sprint/ (accessed Aug. 13, 2022).

[303] G. N. Nedeltcheva and E. Shoikova, 'Coupling design thinking, user experience design and agile: Towards cooperation framework', *ACM International Conference Proceeding Series*, pp. 225–229, Dec. 2017, doi: 10.1145/3175684.3175711.

[304] J. H. J. Yun, K. H. Jung, and T. Yigitcanlar, 'Open Innovation of James Watt and Steve Jobs: Insights for Sustainability of Economic Growth', *Sustainability 2018, Vol. 10, Page 1553*, vol. 10, no. 5, p. 1553, May 2018, doi: 10.3390/SU10051553.

[305] R. G. Cooper, 'The new product process: A decision guide for management', *Journal of Marketing Management*, vol. 3, no. 3, pp. 238–255, Jan. 1988, doi: 10.1080/0267257X.1988.9964044.

[306] S. J. Edgett, 'Idea-to-Launch (Stage-Gate ®) Model: An Overview', *Stage-Gate International*, pp. 1–5, 2015.

[307] R. G. Cooper, 'Stage-gate systems: A new tool for managing new products', *Bus Horiz*, vol. 33, no. 3, pp. 44–54, May 1990, doi: 10.1016/0007-6813(90)90040-I.

[308] L. Petricioli and K. Fertalj, 'Agile Software Development Methods and Hybridization Possibilities Beyond Scrumban', *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, pp. 1093–1098, May 2022, doi: 10.23919/MIPRO55190.2022.9803402.

[309] N. Baruah, A. Buragohain, and S. Bhushan Saikia, 'Agile Team Management: A Review', *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 12, 2016, doi: 10.17148/IJARCCE.2016.51272.

[310] M. Alqudah and R. Razali, 'An empirical study of Scrumban formation based on the selection of scrum and Kanban practices', *Int J Adv Sci Eng Inf Technol*, vol. 8, no. 6, pp. 2315–2322, 2018, doi: 10.18517/IJASEIT.8.6.6566.

[311] R. G. Cooper and A. F. Sommer, 'Agile–Stage-Gate for Manufacturers', *Research-Technology Management*, vol. 61, no. 2, pp. 17–26, Mar. 2018, doi: 10.1080/08956308.2018.1421380.

[312] R. G. Cooper and A. F. Sommer, 'The Agile–Stage-Gate Hybrid Model: A Promising New Approach and a New Research Opportunity', *Journal of Product Innovation Management*, vol. 33, no. 5, pp. 513–526, Sep. 2016, doi: 10.1111/JPIM.12314.

[313] R. G. Cooper and A. F. Sommer, 'Agile-Stage-Gate: New idea-to-launch method for manufactured new products is faster, more responsive', *Industrial Marketing Management*, vol. 59, pp. 167–180, Nov. 2016, doi: 10.1016/J.INDMARMAN.2016.10.006.

[314] A. Eljayar and J. S. Busch, 'Agile-Stage-Gate Approach: Exploratory Research on the Structure, Roles, and Responsibilities', *Athens Journal of Technology and*

Engineering, vol. 8, no. 1, pp. 39–90, 2021, doi: 10.30958/ajte.8-1-3.

[315] R. G. Cooper and A. F. Sommer, 'Agile–Stage-Gate for Manufacturers: Changing the Way New Products Are DevelopedIntegrating Agile project management methods into a Stage-Gate system offers both opportunities and challenges', *Research Technology Management*, vol. 61, no. 2, pp. 17–26, Mar. 2018, doi: 10.1080/08956308.2018.1421380.

[316] R. G. Cooper and A. F. Sommer, 'The Agile–Stage-Gate Hybrid Model: A Promising New Approach and a New Research Opportunity', *Journal of Product Innovation Management*, vol. 33, no. 5, pp. 513–526, Sep. 2016, doi: 10.1111/JPIM.12314.

[317] D. F. O. de Paula and C. C. Araújo, 'Pet empires: Combining design thinking, lean startup and agile to learn from failure and develop a successful game in an undergraduate environment', *Communications in Computer and Information Science*, vol. 617, pp. 30–34, 2016, doi: 10.1007/978-3-319-40548-3_5/FIGURES/1.

[318] F. Dobrigkeit, D. De Paula, and N. Carroll, 'InnoDev Workshop: A One Day Introduction to Combining Design Thinking, Lean Startup and Agile Software Development', *2020 IEEE 32nd Conference on Software Engineering Education and Training, CSEE and T 2020*, pp. 189–198, Nov. 2020, doi: 10.1109/CSEET49119.2020.9206184.

[319] 'Cpputest'. http://cpputest.github.io/ (accessed Dec. 18, 2022).

[320] 'Unity — Throw The Switch'. http://www.throwtheswitch.org/unity/ (accessed Dec. 18, 2022).

[321] L. Rose and S. P. Clarke, 'Vital Signs', *American Journal of Nursing*, vol. 110, no. 5, p. 11, May 2010, doi: 10.1097/01.NAJ.0000372049.58200.DA.

[322] C. Lockwood, T. Conroy-Hiller, and T. Page, 'Vital signs', *JBI Reports*, vol. 2, no. 6, pp. 207–230, Jul. 2004, doi: 10.1111/J.1479-6988.2004.00012.X.

[323] Fortune Business Insights, 'mHealth Market mHealth Market Size, Share & Industry Analysis, By Category (By Apps {Disease & Treatment Management, Wellness Management}, By Wearable {Body & Temperature Monitors, Glucose Monitors})', 2020.

[324] C. H. Vinkers *et al.*, 'The effect of stress on core and peripheral body temperature in humans', *The International Journal on the Biology of Stress*, vol. 16, no. 5, pp. 520–530, Sep. 2013, doi: 10.3109/10253890.2013.807243.

[325] B. Coiffard, A. B. Diallo, S. Mezouar, M. Leone, and J. L. Mege, 'A Tangled Threesome: Circadian Rhythm, Body Temperature Variations, and the Immune System', *Biology 2021, Vol. 10, Page 65*, vol. 10, no. 1, p. 65, Jan. 2021, doi: 10.3390/BIOLOGY10010065.

[326] A. Sixsmith, N. Johnson, and R. Whatmore, 'Pyroelectric IR sensor arrays for fall detection in the older population', *Journal De Physique. IV : JP*, vol. 128, pp. 153–160, Sep. 2005, doi: 10.1051/JP4:2005128024.

[327] D. W. Marion, 'Therapeutic Moderate Hypothermia and Fever', *Cerebral Blood Flow*, pp. 272–276, 2003, doi: 10.1007/978-3-642-56036-1_20.

[328] D. H. Spodick, P. Raju, R. L. Bishop, and R. D. Rifkin, 'Operational definition of normal sinus heart rate', *Am J Cardiol*, vol. 69, no. 14, pp. 1245–1246, May 1992, doi: 10.1016/0002-9149(92)90947-W.

[329] W. H. Organization, 'Hypertension'. 2022.

[330] P. B. Lovett, J. M. Buchwald, K. Stürmann, and P. Bijur, 'The vexatious vital: Neither clinical measurements by nurses nor an electronic monitor provides accurate measurements of respiratory rate in triage', *Ann Emerg Med*, vol. 45, no. 1, pp.

68–76, Jan. 2005, doi: 10.1016/J.ANNEMERGMED.2004.06.016.

[331] R. Subhi, K. Smith, and T. Duke, 'When should oxygen be given to children at high altitude? A systematic review to define altitude-specific hypoxaemia', *Arch Dis Child*, vol. 94, no. 1, pp. 6–10, Jan. 2009, doi: 10.1136/ADC.2008.138362.

[332] J. Gunst and G. Van den Berghe, 'Blood Glucose Control in the Intensive Care Unit: Benefits and Risks', *Semin Dial*, vol. 23, no. 2, pp. 157–162, Mar. 2010, doi: 10.1111/J.1525-139X.2010.00702.X.

[333] H. Rodríguez, M. Williams, J. S. Wilkes, and R. D. Rogers, 'Ionic liquids for liquid-in-glass thermometers', *Green Chemistry*, vol. 10, no. 5, pp. 501–507, May 2008, doi: 10.1039/B800366A.

[334] Roya. Habibian, Shahriar. Salehi, Reza. Imani, Bahman. Sadeghi, and Khadijeh. Hatamipou, 'Comparative study of measuring body temperature by mercury and digital thermometer', *Iran Journal of Nursing*, 2009.

[335] S. Khan *et al.*, 'Comparative accuracy testing of non-contact infrared thermometers and temporal artery thermometers in an adult hospital setting', *Am J Infect Control*, vol. 49, no. 5, pp. 597–602, May 2021, doi: 10.1016/J.AJIC.2020.09.012.

[336] A. Bansal and R. Joshi, 'Portable out-of-hospital electrocardiography: A review of current technologies', *J Arrhythm*, vol. 34, no. 2, pp. 129–138, Apr. 2018, doi: 10.1002/JOA3.12035.

[337] J. Parak and J. Havlik, 'ECG signal processing and heart rate frequency detection methods', in *Proceedings of Technical Computing Prague*, Prague, 2011.

[338] N. Saquib, M. T. I. Papon, I. Ahmad, and A. Rahman, 'Measurement of heart rate using photoplethysmography', *Proceedings of 2015 International Conference on Networking Systems and Security, NSysS 2015*, Feb. 2015, doi: 10.1109/NSYSS.2015.7043525.

[339] C. F. Babbs, 'The origin of Korotkoff sounds and the accuracy of auscultatory blood pressure measurements', *Journal of the American Society of Hypertension*, vol. 9, no. 12, pp. 935-950.e3, Dec. 2015, doi: 10.1016/J.JASH.2015.09.011.

[340] G. Beevers, G. Y. H. Lip, and E. O'brien, 'Blood pressure measurement', *BMJ*, vol. 322, no. 7293, pp. 1043–1047, Apr. 2001, doi: 10.1136/BMJ.322.7293.1043.

[341] M. Mostafa, A. M. Hasanin, O. S. Elsayed, M. M. Mostafa, and K. Sarhan, 'Accuracy of oscillometric blood pressure measurement at both arms in the lateral position', *Blood Press Monit*, pp. 364–372, 2021, doi: 10.1097/MBP.0000000000000546.

[342] J. Nedoma *et al.*, 'Magnetic Resonance Imaging Compatible Non-Invasive Fibre-Optic Sensors Based on the Bragg Gratings and Interferometers in the Application of Monitoring Heart and Respiration Rate of the Human Body: A Comparative Study', *Sensors 2018, Vol. 18, Page 3713*, vol. 18, no. 11, p. 3713, Oct. 2018, doi: 10.3390/S18113713.

[343] C. Massaroni, A. Nicolò, M. Sacchetti, and E. Schena, 'Contactless Methods For Measuring Respiratory Rate: A Review', *IEEE Sens J*, vol. 21, no. 11, pp. 12821–12839, 2021, doi: 10.1109/JSEN.2020.3023486.

[344] P. H. Charlton *et al.*, 'An impedance pneumography signal quality index: Design, assessment and application to respiratory rate monitoring', *Biomed Signal Process Control*, vol. 65, p. 102339, Mar. 2021, doi: 10.1016/J.BSPC.2020.102339.

[345] M. R. Miller *et al.*, 'Standardisation of spirometry', *European Respiratory Journal*, vol. 26, no. 2, pp. 319–338, 2005, doi: 10.1183/09031936.05.00034805.

[346] S. D. Bergese *et al.*, 'Multicenter Study Validating Accuracy of a Continuous Respiratory Rate Measurement Derived From Pulse Oximetry: A Comparison With Capnography', *Anesth Analg*, vol. 124, no. 4, p. 1153, 2017, doi: 10.1213/ANE.0000000000001852.

[347] A. Moço and W. Verkruysse, 'Pulse oximetry based on photoplethysmography imaging with red and green light: Calibratability and challenges', *J Clin Monit Comput*, vol. 35, no. 1, pp. 123–133, Feb. 2021, doi: 10.1007/S10877-019-00449-Y/FIGURES/11.

[348] J. Wang, 'Electrochemical glucose biosensors', *Chem Rev*, vol. 108, no. 2, pp. 814–825, Feb. 2008, doi: 10.1021/CR068123A/ASSET/CR068123A.FP.PNG_V03.

[349] C. J. D. McKinlay, J. G. Chase, J. Dickson, D. L. Harris, J. M. Alsweiler, and J. E. Harding, 'Continuous glucose monitoring in neonates: a review', *Maternal Health, Neonatology and Perinatology 2017 3:1*, vol. 3, no. 1, pp. 1–9, Oct. 2017, doi: 10.1186/S40748-017-0055-Z.

[350] D. Sudana and A. W. R. Emanuel, 'How Big Data in Health 4.0 Helps Prevent the Spread of Tuberculosis', *Proceedings - 2019 2nd International Conference on Bioinformatics, Biotechnology and Biomedical Engineering - Bioinformatics and Biomedical Engineering, BioMIC 2019*, Sep. 2019, doi: 10.1109/BIOMIC48413.2019.9034696.

[351] H. Li, A. Shrestha, H. Heidari, J. Le Kernec, and F. Fioranelli, 'A Multisensory Approach for Remote Health Monitoring of Older People', *IEEE J Electromagn RF Microw Med Biol*, vol. 2, no. 2, pp. 102–108, Jun. 2018, doi: 10.1109/JERM.2018.2827099.

[352] L. García, J. Tomás, L. Parra, and J. Lloret, 'An m-health application for cerebral stroke detection and monitoring using cloud services', *Int J Inf Manage*, vol. 45, pp. 319–327, Apr. 2019, doi: 10.1016/J.IJINFOMGT.2018.06.004.

[353] M. H. Lin, W. L. Yuan, T. C. Huang, H. F. Zhang, J. T. Mai, and J. F. Wang, 'Clinical effectiveness of telemedicine for chronic heart failure: a systematic review and meta-analysis', *J Investig Med*, vol. 65, no. 5, pp. 899–911, Jun. 2017, doi: 10.1136/JIM-2016-000199.

[354] P. Rashidi and A. Mihailidis, 'A survey on ambient-assisted living tools for older adults', *IEEE J Biomed Health Inform*,

vol. 17, no. 3, pp. 579–590, 2013, doi: 10.1109/JBHI.2012.2234129.

[355] N. S. Shivakumar and M. Sasikala, 'Design of vital sign monitor based on wireless sensor networks and telemedicine technology', *Proceeding of the IEEE International Conference on Green Computing, Communication and Electrical Engineering, ICGCCEE 2014*, Oct. 2014, doi: 10.1109/ICGCCEE.2014.6922257.

[356] N. Isakadze and S. S. Martin, 'How useful is the smartwatch ECG?', *Trends Cardiovasc Med*, vol. 30, no. 7, pp. 442–448, Oct. 2020, doi: 10.1016/J.TCM.2019.10.010.

[357] C. De Asmundis *et al.*, 'Comparison of the patient-activated event recording system vs. traditional 24 h Holter electrocardiography in individuals with paroxysmal palpitations or dizziness', *EP Europace*, vol. 16, no. 8, pp. 1231–1235, Aug. 2014, doi: 10.1093/EUROPACE/EUT411.

[358] Z. Liang and M. A. Chapa-Martell, 'Validation of Omron Wearable Blood Pressure Monitor HeartGuideTM in Free-Living Environments', *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 362 LNICST, pp. 339–350, 2021, doi: 10.1007/978-3-030-70569-5_22/TABLES/2.

[359] Y.-T. Kwak, Y. Lee, and Y. You, 'Statistical Estimation of Body Temperature from Skin Temperature for Smart Band', in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, 2019, pp. 1477–1481. doi: 10.1109/ICTC46691.2019.8939975.

[360] M. Popescu and A. Mahnot, 'Early illness recognition using in-home monitoring sensors and multiple instance learning', *Methods Inf Med*, vol. 51, no. 4, pp. 359–367, 2012, doi: 10.3414/ME11-02-0042.

[361] L. Klack, T. Schmitz-Rode, W. Wilkowska, K. Kasugai, F. Heidrich, and M. Ziefle, 'Integrated home monitoring and compliance optimization for patients with mechanical

circulatory support devices', *Ann Biomed Eng*, vol. 39, no. 12, pp. 2911–2921, Dec. 2011, doi: 10.1007/S10439-011-0407-1.

[362] S. L. Grace *et al.*, 'Perceptions of seniors with heart failure regarding autonomous zero-effort monitoring of physiological parameters in the smart-home environment', *Heart & Lung*, vol. 46, no. 4, pp. 313–319, Jul. 2017, doi: 10.1016/J.HRTLNG.2017.04.007.

[363] K. Avgerinakis, A. Briassouli, and I. Kompatsiaris, 'Recognition of activities of daily living for smart home environments', *Proceedings - 9th International Conference on Intelligent Environments, IE 2013*, pp. 173–180, 2013, doi: 10.1109/IE.2013.37.

[364] P. S. Pandian *et al.*, 'Smart Vest: Wearable multi-parameter remote physiological monitoring system', *Med Eng Phys*, vol. 30, no. 4, pp. 466–477, May 2008, doi: 10.1016/J.MEDENGPHY.2007.05.014.

[365] E. Sardini and M. Serpelloni, 'T-shirt for vital parameter monitoring', *Lecture Notes in Electrical Engineering*, vol. 162 LNEE, pp. 201–205, 2014, doi: 10.1007/978-1-4614-3860-1_35/FIGURES/00356.

[366] L. Pigini *et al.*, 'Pilot Test of a New Personal Health System Integrating Environmental and Wearable Sensors for Telemonitoring and Care of Elderly People at Home (SMARTA Project)', *Gerontology*, vol. 63, no. 3, pp. 281–286, Apr. 2017, doi: 10.1159/000455168.

[367] M. Pham, Y. Mengistu, H. M. Do, and W. Sheng, 'Cloud-Based Smart Home Environment (CoSHE) for home healthcare', *IEEE International Conference on Automation Science and Engineering*, vol. 2016-Novem, pp. 483–488, Nov. 2016, doi: 10.1109/COASE.2016.7743444.

[368] R. Villar, T. Beltrame, and R. L. Hughson, 'Validation of the Hexoskin wearable vest during lying, sitting, standing, and walking activities', *https://doi.org/10.1139/apnm-2015-0140*, vol. 40, no. 10, pp. 1019–1024, Sep. 2015, doi: 10.1139/APNM-2015-0140.

[369] T. Li, S. Divatia, J. McKittrick, J. Moss, N. M. Hijnen, and L. B. Becker, 'A pilot study of respiratory rate derived from a wearable biosensor compared with capnography in emergency department patients', *Open Access Emerg Med*, vol. 11, p. 103, 2019, doi: 10.2147/OAEM.S198842.

[370] F. J. Richmond *et al.*, 'A novel maturity index for assessing medical device startups', *J Clin Transl Sci*, vol. 6, no. 1, p. 9, Aug. 2022, doi: 10.1017/CTS.2022.436.

[371] M. Gutbrod, J. Münch, and M. Tichy, 'How do software startups approach experimentation? Empirical results from a qualitative interview study', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10611 LNCS, pp. 297–304, 2017, doi: 10.1007/978-3-319-69926-4_21/TABLES/2.

[372] E. S. Rasmussen and S. Tanev, 'Lean start-up: Making the start-up more successful', *Start-Up Creation: The Smart Eco-Efficient Built Environment*, pp. 39–56, Jan. 2016, doi: 10.1016/B978-0-08-100546-0.00003-0.

[373] J. Melegati, R. Chanin, A. Sales, R. Prikladnicki, and X. Wang, 'MVP and experimentation in software startups: A qualitative survey', *Proceedings - 46th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2020*, pp. 322–325, Aug. 2020, doi: 10.1109/SEAA51224.2020.00060.

[374] H. Edison, N. M. Smørsgård, X. Wang, and P. Abrahamsson, 'Lean Internal Startups for Software Product Innovation in Large Companies: Enablers and Inhibitors', *Journal of Systems and Software*, vol. 135, pp. 69–87, Jan. 2018, doi: 10.1016/J.JSS.2017.09.034.

[375] D. Dennehy, L. Kasraian, P. O'Raghallaigh, K. Conboy, D. Sammon, and P. Lynch, 'A Lean Start-up approach for developing minimum viable products in an established company', *https://doi.org/10.1080/12460125.2019.1642081*, vol. 28, no. 3, pp. 224–232, Jul. 2019, doi: 10.1080/12460125.2019.1642081.

[376] M. T. Sharma, 'COMPARISON BETWEEN AGILE AND HEAVYWEIGHT METHODOLOGY: AN ANALYSIS', *International Journal of Technical Research and Applications*, vol. 4, no. 1, pp. 227–233.

[377] P. Serrador and J. K. Pinto, 'Does Agile work? — A quantitative analysis of agile project success', *International Journal of Project Management*, vol. 33, no. 5, pp. 1040–1051, Jul. 2015, doi: 10.1016/J.IJPROMAN.2015.01.006.

[378] S. Overhage and S. Schlauderer, 'Investigating the long-term acceptance of agile methodologies: An empirical study of developer perceptions in Scrum projects', *Proceedings of the Annual Hawaii International Conference on System Sciences*, pp. 5452–5461, 2012, doi: 10.1109/HICSS.2012.387.

[379] H. Lei, F. Ganjeizadeh, P. K. Jayachandran, and P. Ozcan, 'A statistical analysis of the effects of Scrum and Kanban on software development projects', *Robot Comput Integr Manuf*, vol. 43, pp. 59–67, Feb. 2017, doi: 10.1016/J.RCIM.2015.12.001.

[380] X. Wang, L. Zhao, Y. Wang, and J. Sun, 'The Role of Requirements Engineering Practices in Agile Development: An Empirical Study', *Communications in Computer and Information Science*, vol. 432 CCIS, pp. 195–205, 2014, doi: 10.1007/978-3-662-43610-3_15/COVER.

[381] F. McCaffery, K. Trektere, and O. Ozcan-Top, 'Agile – is it suitable for medical device software development?', *Communications in Computer and Information Science*, vol. 609, pp. 417–422, 2016, doi: 10.1007/978-3-319-38980-6_30/COVER.

[382] M. McHugh, F. McCaffery, and V. Casey, 'Barriers to Using Agile Software Development Practices within the Medical Device Industry', *Conference papers*, Jun. 2012, doi: https://doi.org/10.21427/hq34-mh45.