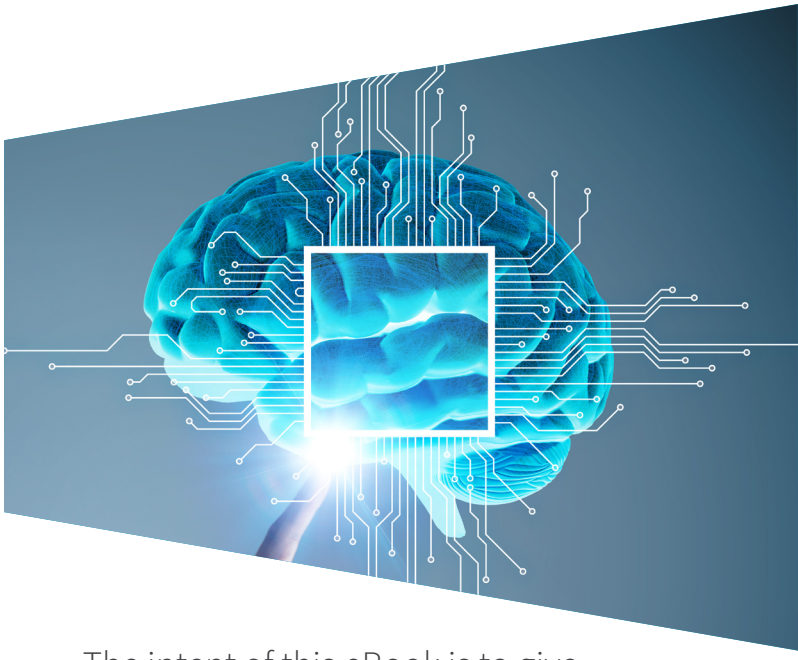
ASPHALION

Ins and outs of software *as a medical device*

Key technology and regulatory aspects to consider in MDSW life cycle management

Contents

1. Introduction



Digital health technologies are being developed into many kinds of applications that are already changing therapeutic and diagnostic approaches to disease management and will certainly dramatically change medicine in the coming years. This include software platforms that enable better doctor-to-patient communication and clinical history records management tools, software modules that drive medical electric instruments and stand-alone software solutions that process medical information against big data stored in the cloud, etc. **Algorithms** can automat the analysis of large datasets, speeding up patient management decisions. Software technologies can combine input **data** of different natures and unveil clinically relevant information. **Software, in short, has enormous potential in health management and it is crucial that digital health application developers understand the underlying regulatory framework that both their organisations and these appliances need to comply with for the legal placing on the EU market.**

The intent of this eBook is to give an overview about the **guidelines of developing medical device software, which in Europe is a particular software category** that is governed by either **Regulation EU/2017/745 on medical devices (MDR),** or **Regulation EU/2017/746 on in vitro diagnostic medical devices (IVDR).**

The content shared in the current document is based on **Regulation EU/2017/745 on medical devices (MDR), Regulation EU/2017/746, IEC62304:2006** as well as **ISO13485:2016 and ISO14971:2019;** and summarises a framework of life cycle processes with activities and tasks required to fulfil for the safe design of medical device software.

2. Software developments *in the scope of EU medical device regulation*

Medical device software requires safe and effective implementation, a clear knowledge of what the software is intended to do and demonstration that the use of it fulfils those intentions without causing any unacceptable risk.

If you're not sure if your product falls into the **MD or IVD product category,** check the Add-on to find out.

So, when software products fall into the **MD or IVD product category,** compliance with either the MDR or the company in the market that will be responsible for the product, and which shall be identified as "manufacturer" in the software labelling.

In both cases (MDSW or IVDSW), compliance with MDR and IVDR, respectively, needs to be demonstrated based on compliance of the product with the correspondingly applicable general safety and performance requirements (GSPRs), as laid down in Annex I of both regulations. **SW-specific GSPRs** are provided in points 17 and 16 of Annex I in MDR

Regulatory *add-on*

MDSW or IVDSW Qualification:
'Medical device' means 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.

and IVDR, respectively, which refer to Electronic programmable systems. Hereinafter, we will refer to this kind of SW as MDSW, for short, regardless of whether it is MDSW or IVDSW. These requirements demand that MDSW operates in a repeatable and reliable way, and that it performs as intended and claimed by the manufacturer in the product labelling (About this software...). Other requirements include aspects such as **interoperability with mobile devices and platforms, IT networks and cybersecurity.** Furthermore, these two points of Annex I on MDR and IVDR require that compliance of the MDSW to these aspects is demonstrated by the manufacturer considering both normal and single-fault conditions, implying that appropriate control measures must be in place to effectively prevent or mitigate as much as possible any risk of such single-fault conditions, including harmful consequences of lack of performance of the **MDSW**.

Regulatory *add-on*

'In vitro diagnostic medical device' means 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:

(a) concerning a physiological or pathological process or state;

(b) concerning congenital physical or mental impairments;

(c) concerning the predisposition to a medical condition or a disease;

(d) to determine the safety and compatibility with potential recipients;

(e) to predict treatment response or reactions;

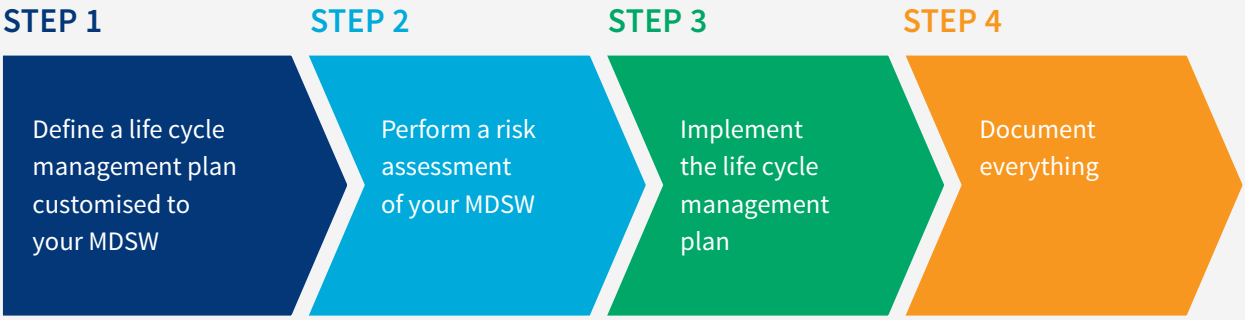
(f) to define or monitor therapeutic measures.

3. Technical standards *in MDSW development and their interrelation*

In regulation, requirements as defined in the legal documents (regulations, directives, their amendments and correspondingly implemented or transposed national documents) are often difficult to implement and manufacturers need to refer to interpretative documents to be able to define an effective compliance strategy. In MD regulation, this help is provided by the guidelines, technical reports and technical standards. These documents, published by organisations such as the **European Commission Medical Device Coordination Group (MDCG),** the **International Medical Device Regulators Forum (IMDRF),** the **International Standardisation Organisation (ISO),** and the **European Committee for Standardization / European Committee for Electrotechnical Standardization (CEN/CENELEC),** among others, are recognised by authorities and notified bodies as best practice and state-of-the-art methods and approaches to **design, manufacturing, verification and maintenance aspects of medical devices** (both generally and product-specific), **which are really useful to demonstrate conformity of the products to either MDR or IVDR for CE certification.**

The not so good news is that the list of such interpretative documents that could be helpful to define an approach for demonstrating compliance of MDSW with the applicable GSPRs could be quite long; thus, having to deal with this bunch of complex technical literature is a huge task. As a start, however, we would recommend **MDSW developers to become familiar with the following documents**: standards IEC 62304

Medical device software - Software life cycle processes, IEC 60601-1-4 Medical electrical equipment - Part 1-4: General requirements for safety - Collateral standard: Programmable electrical medical systems and ISO 14971 Medical devices — Application of risk management to medical devices, IEC 82304-1:2016 Health software — Part 1: General requirements for product safety and technical report IEC/TR 80002-1 Medical device software — Part 1: Guidance on the application of ISO 14971 to medical device software. In the following, we will try to bring some high-level clarity on how to apply these standards in the context of MDSW development for compliance with the SW-specific GSPRs mentioned above.



Step 1: Define a life cycle management plan customised to your MDSW

First of all, we need to understand what **IEC 62304 is useful for: it serves the purpose of defining and executing a plan whereby the MDSW will be developed, verified and validated at SW system level up to release, and then maintained along its life cycle, every time changes need to be implemented due to either problems that have been encountered upon use in real context or due to updates for improved performance.** Thus, IEC 62304 will not cover the part of the clinical validation of the SW system as a medical device, but the whole of its technical, non-clinical part, from release to maintenance and decommissioning.

Next, we need to determine which **risk category** our SW system falls into, according to the basic question of whether a failure of the SW can cause a hazardous situation that would lead to unacceptable risk, understood as either serious or non-serious injury to the operator or to the patient. To answer this question, risk mitigation

measures can be considered but, only those that are external to the SW system (either hardware-related or software-related, but coming from an independent SW system). If the answer is no, then the risk class of the **SW system is A** and the life cycle management plan can be kept to a minimum, whereby only a few clauses of the standard and a limited level of detail will need to be addressed in the plan. If the answer is that a failure mode can indeed lead to non-serious injury, then **the risk class is B** and the life cycle management plan needs to address more clauses of the standard and level of detail will need to be higher. If the answer is that a **failure mode can lead to a serious injury or death of the patient,** then the plan must cover all the clauses and the level of detail must be exhaustive. **Since SW systems may be integrated by various modules (or items), this analysis can be applied to each module independently, and must be reassessed upon every change to the MDSW.**

Step 2: Perform a risk assessment of your MDSW

Another important aspect of **IEC 62304 is risk management, especially for risk classes B and C, and should be at the basis of the whole life cycle management process. Even though this is an aspect of the standard that would in principle not apply to SW of risk-class A,** from an MDR point of view, risk management is of utmost importance in any MDSW case and therefore cannot be skipped. In MD regulation, risk management shall be performed according to standard ISO 14971, and therefore, the two standards need to be applied in conjunction along any MDSW development. According to ISO 14971, risk management should be planned including a definition and description of the MDSW, its use conditions and potential sources of hazardous situations; risk acceptability criteria should be defined and justified, and risk analysis performed, applying any suitable method (fault tree analysis or FTA; failure mode effects and criticality analysis or FEMCA; hazard and operability or HAZOP; etc.), and the whole process summarised in a risk management report that ends up discussing the risk/benefit profile of the MDSW.

Step 3: Implement the life cycle management plan

From this point onwards, the standard can be applied, starting by defining the high-level and low-level architecture of the SW system, defining SW-specific re-

quirements (SSR), associating verification tests for each SSR and establishing a relationship between them and the so-called user-specific requirements of the MDSW (this is what the MDSW is intended to do, by whom and in what clinical context).

Upon implementing the SW architecture and verifying the requirements at unit and integration levels, the whole process can then be validated and the corresponding SW version, released. It is worth mentioning that according to the standard, the validation of the SW system should be executed externally to the SW development team. And as referred to above, the SW release at the technical level does not include the clinical validation of the MDSW in its intended clinical context of use.

Step 4: Document everything

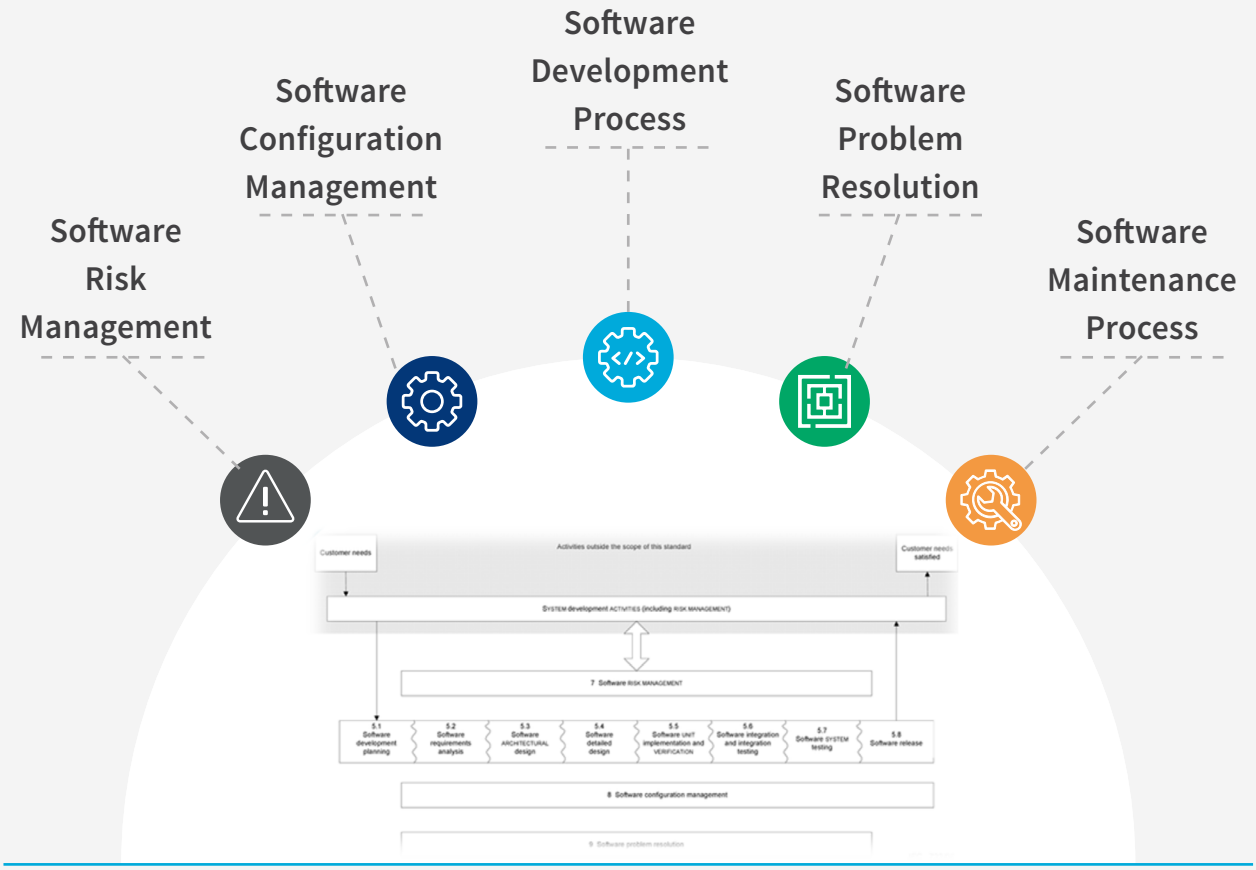
From the perspective of ultimately **CE certifying the MDSW**, it is important that everything that will be done in SW development, **technical validation and risk management** is adequately documented in accordance with the developer / manufacturer **quality management system (QMS).** Therefore, all plans, records and reports must be traceable in the organisation's document archive system, following adequate document control procedures. In this way, the MDSW risk management and the SW system life cycle management processes will produce the required evidence for the later compilation of the **technical documentation (TD)**, which is a key element in the conformity evaluation procedure to be undertaken for **CE certification of the product as a medical device under MDR or IVDR.** More specifically, these documents would be summarised and referred to in sections 4. General Safety and Performance Requirements, 5. Benefit-risk analysis and risk management, and 6. Product verification and validation, subsection 6.1 Preclinical and clinical data (software verification and validation) of the TD, according to Annex II of the MDR.

Regulatory *add-on*

A Quality Management System (QMS) is a formal system that documents the structure, processes, roles, responsibilities and procedures required to achieve effective quality policies and objectives.

4. SW Development *for Medical Device Life Cycle in a nutshell*

The life cycle of medical device software shall include all the following processes:



Usually, these five processes are clearly defined in the designer company's **Quality Management System (QMS),** as part of, the **Software development Life Cycle Standard Operating Procedure (SOP).**

Starting with the **Software Risk Management,** which is applied in accordance with **ISO 14971,** it is initially used to determine the software safety class according to the possible effects resulting from a hazard to which the software system can contribute, by following the steps described in the figure below:

	Severity				
Probability	1	2	3	4	5
1	ACC	ACC	ACC	ACC*	ACC*
2	ACC	ACC	ACC*	ACC*	N ACC
3	ACC	ACC*	ACC*	N ACC	N ACC
4	ACC*	ACC*	N ACC	N ACC	N ACC
5	ACC*	N ACC	N ACC	N ACC	N ACC

The Software Risk Plan document will contain all the related information as well as the actions/mitigations taken to deal with the hazards identified and the risks linked to them.

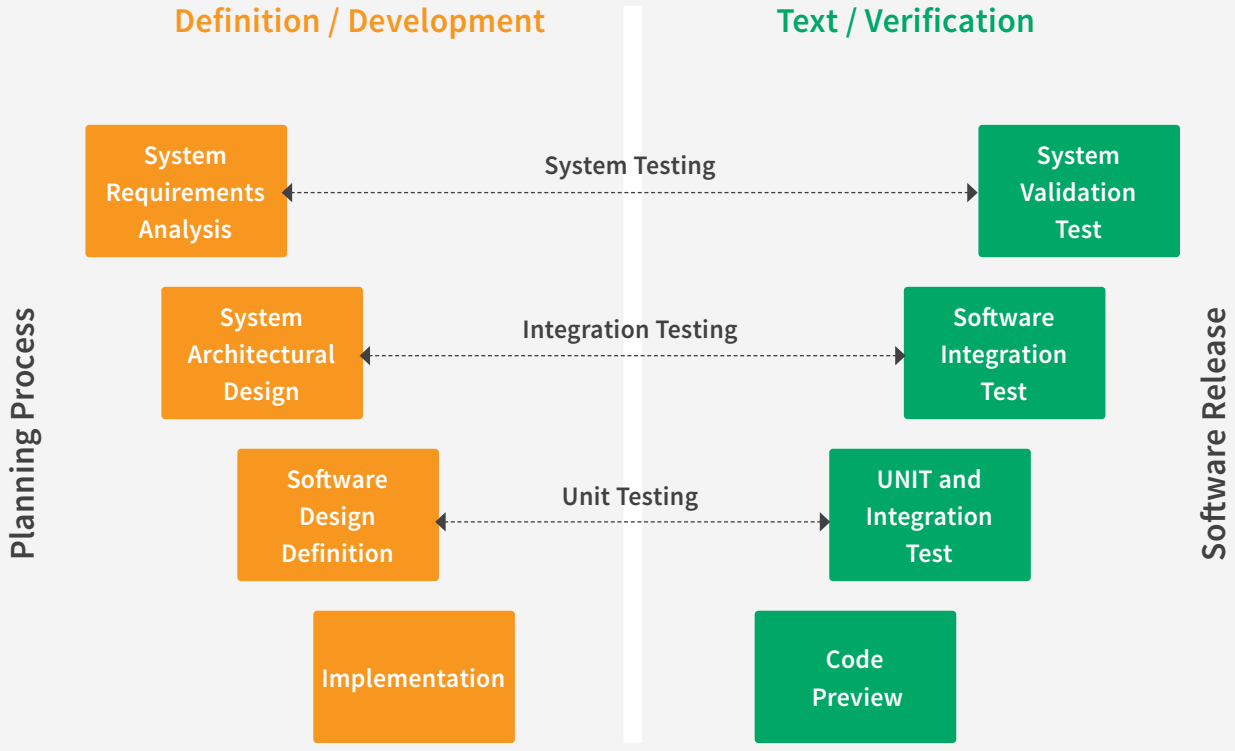
Regulatory *add-on*

Sometimes applying standards to a software product is not that easy, in this case you can use the IEC/TR 80002-1 Medical device software — Part 1: Guidance on the application of ISO 14971 to medical device software as help in your ISO 14971 Risk Management process.

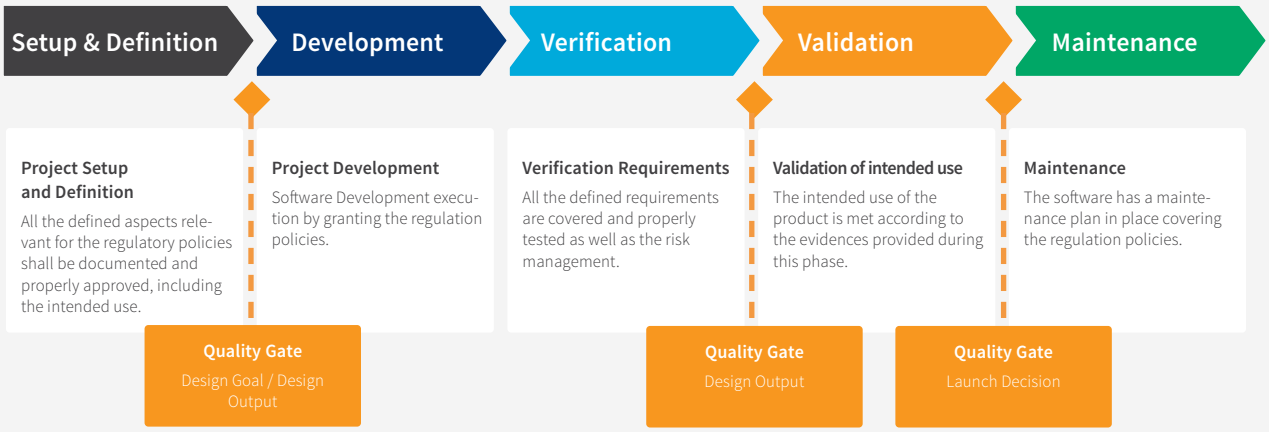
The **Software Configuration Management Process** applies administrative and technical procedures throughout the software life cycle to identify and define Software items (including documentation), Control modifications and releases of items, document and report status of items and change requests, including

- **Configuration identification:** Identify each Software Configuration item and their versions.
- **Change Control:** Including the change request process.
- **Configuration Status** to maintain records of the history of the configuration items.

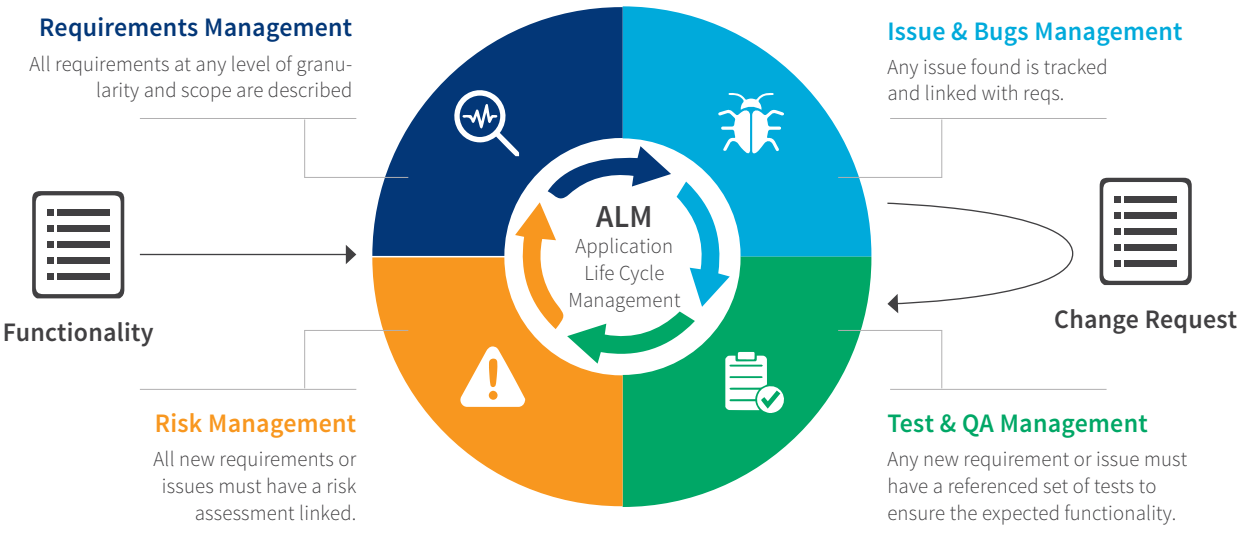
The main process included in the framework is the **Software Development Process,** which defines all the activities and tasks to fulfil to develop the medical device software. For each activity defined in the software development process, and according to the **Software Risk Class** which has been identified, a set of deliverables must be released following a verification process. The verification process is done according to the milestones (gateways) defined in the **PEMS Development life cycle.** One way to understand all these activities is by using the V-Model:



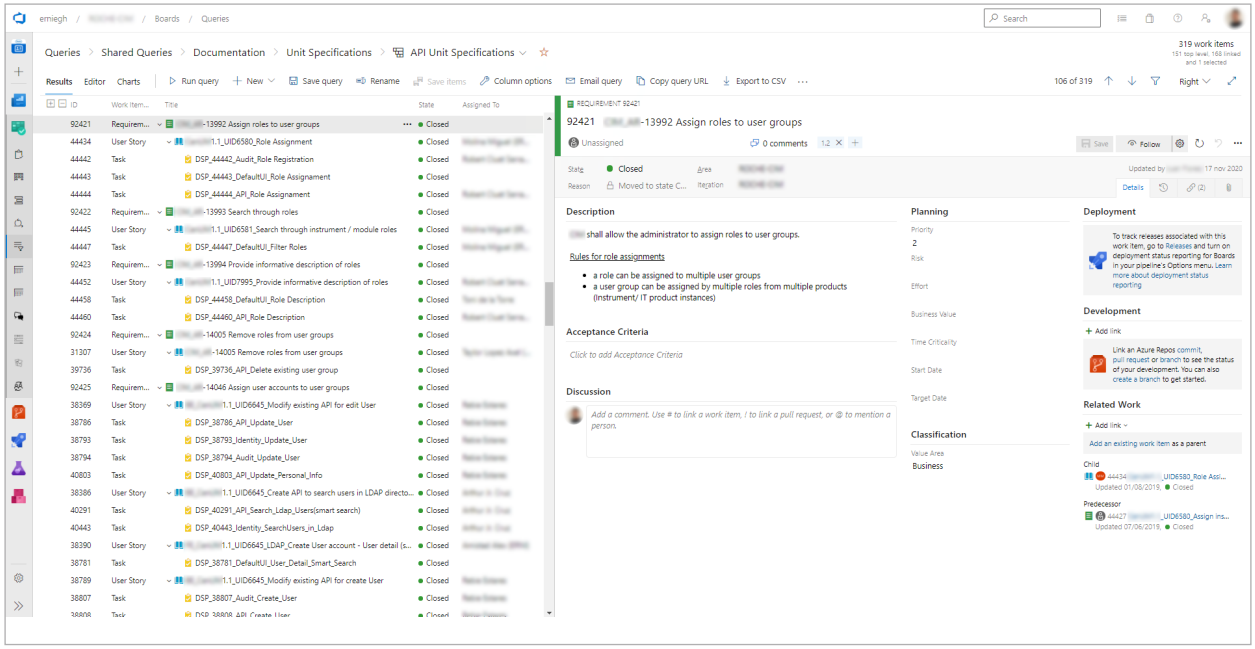
As shown in the figure, it traceability of all the requirements (at all levels), their implementation and their verification activities shall be fulfilled. One possible way to practically implement this model is by using **Quality Gates** as milestones to determine that a set of activities are already completed:



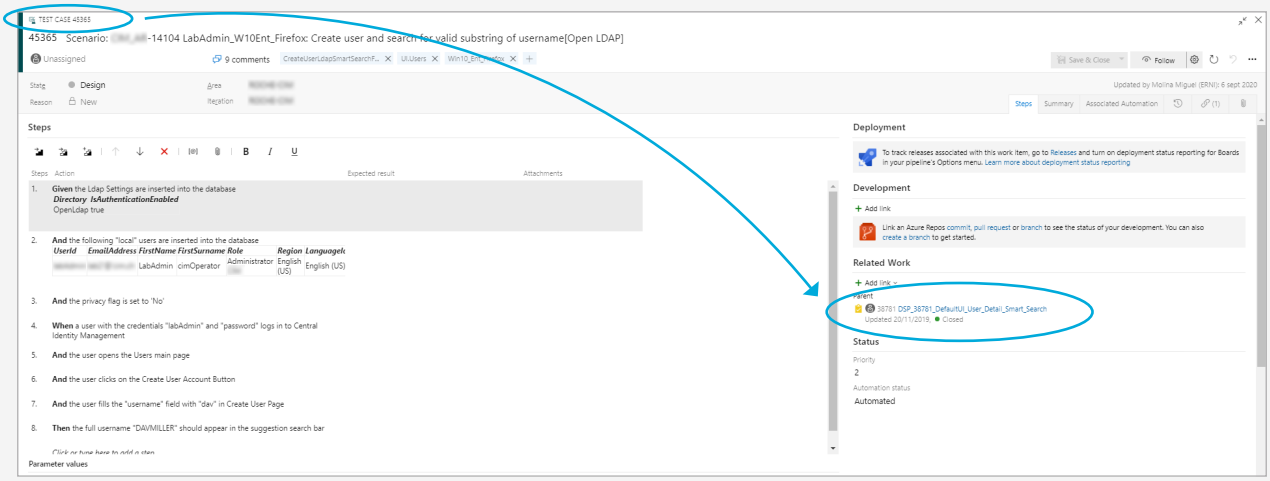
At each Quality Gate, a set of deliverables are defined to be released. The release process includes one author for each deliverable and one or more reviewers and approvers as well as a defined process to create and properly release the documentation. One of the big recommendations that are given when developing medical device software is the usage of an **ALM tool** to synchronise requirements, testing, **risk management** and problem resolution processes.



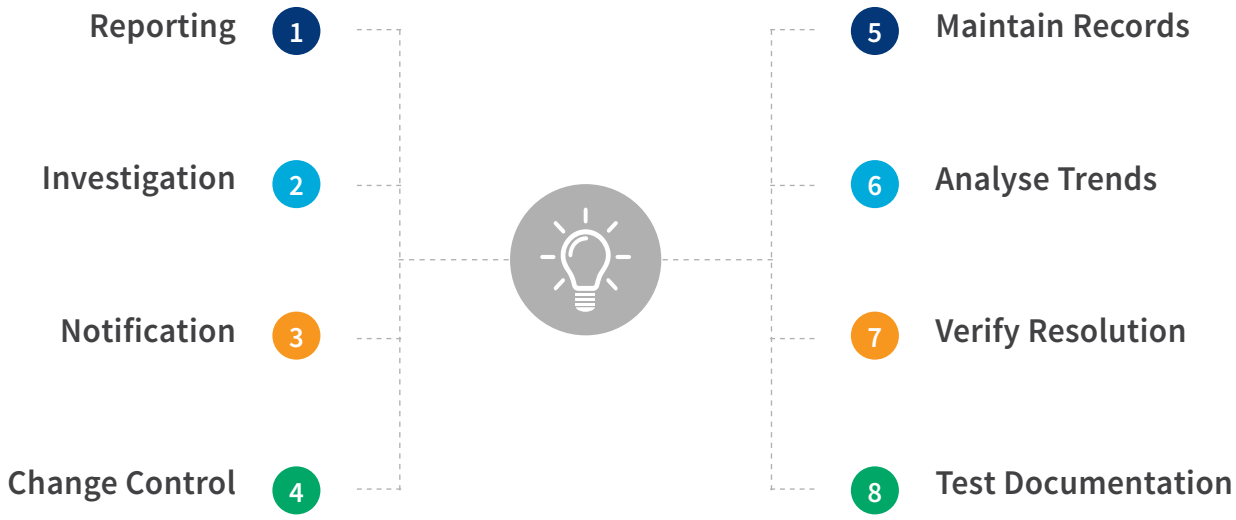
A practical example of the **traceability** between all level requirements is shown in the figure below:



In this particular case, the ALM tool used was able to provide requirements management as well as testing management. Therefore, the **traceability** between the test case and the requirement could be configured on it, the main benefit of which being the possibility to automate part of the documentation process.



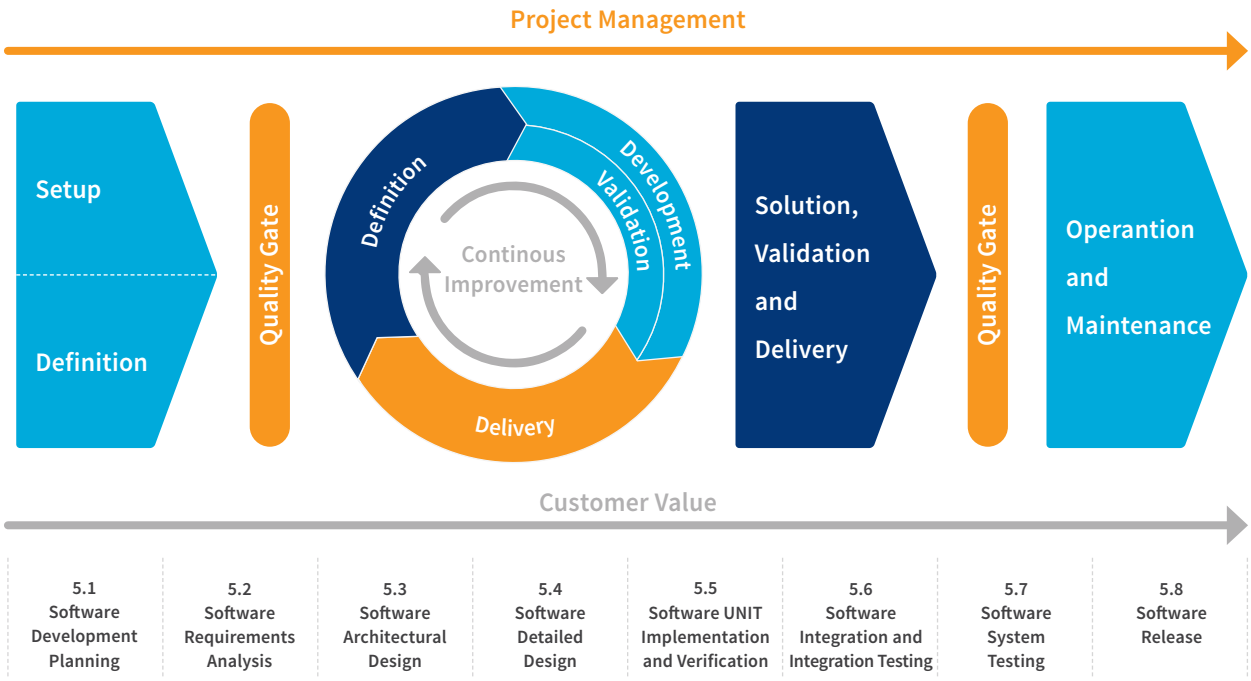
Finally, the **Software Problem Resolution Process** is defined for analysing and resolving the problems discovered during the execution of development, maintenance or other processes, providing a timely, responsible and documented means to ensure the problem is analysed and resolved. A set of mandatory activities shall be fulfilled and documented properly for each problem detected.



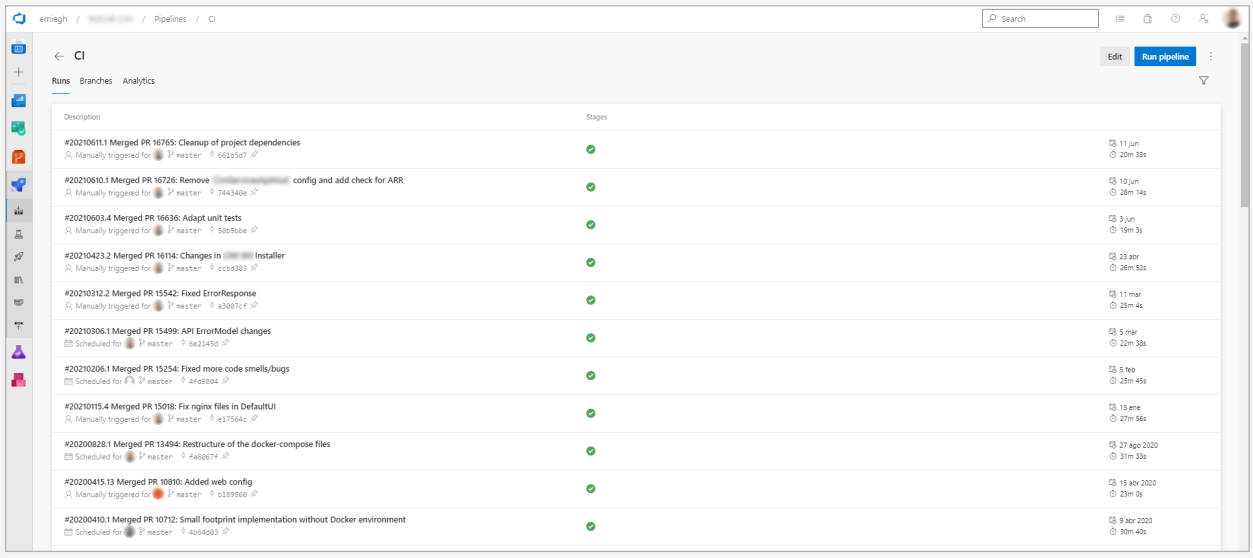
Applying Agile in regulated environments

The application of **Agile frameworks** in **medical device software development** may require a dedicated eBook for sufficient explanation. It is not only about the practical implementation of agile frameworks in the regulation, it is as well about the usage of DevOps practices to apply, for example, continuous integration as well as good branching strategies in the development.

In terms of **Agile Frameworks** is possible to use it between milestones or quality gates as shown in the example below:

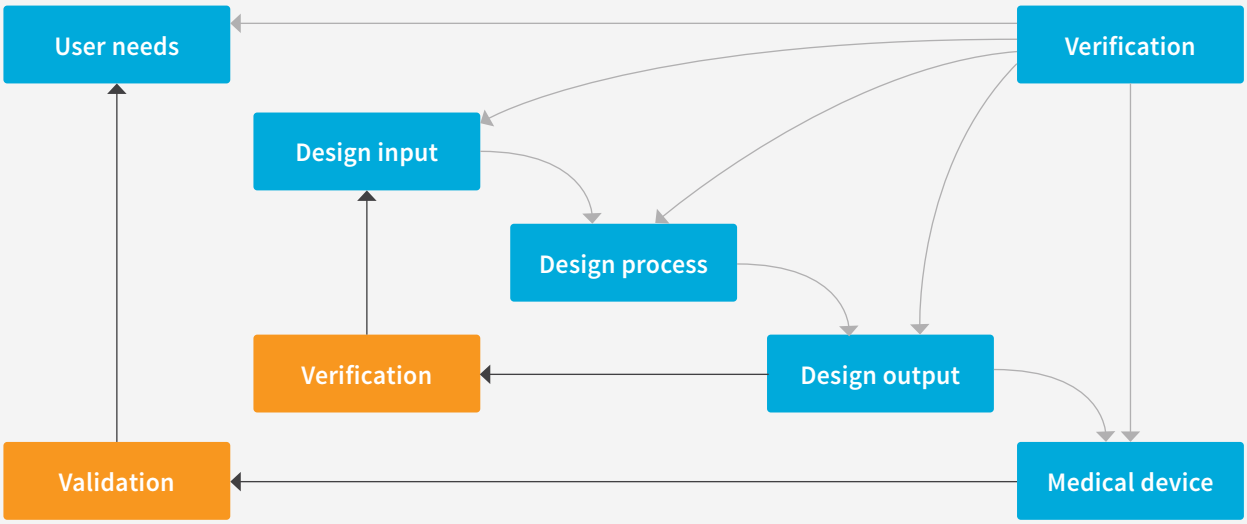


Finally, to integrate **DevOps practices** to boost the **Agile frameworks** used and the development tools which could be used for that intend, there are a separated workshop which could be done and cover from the branching strategy, the configuration management and the continuous integration approach, as it is shown in the figure below:

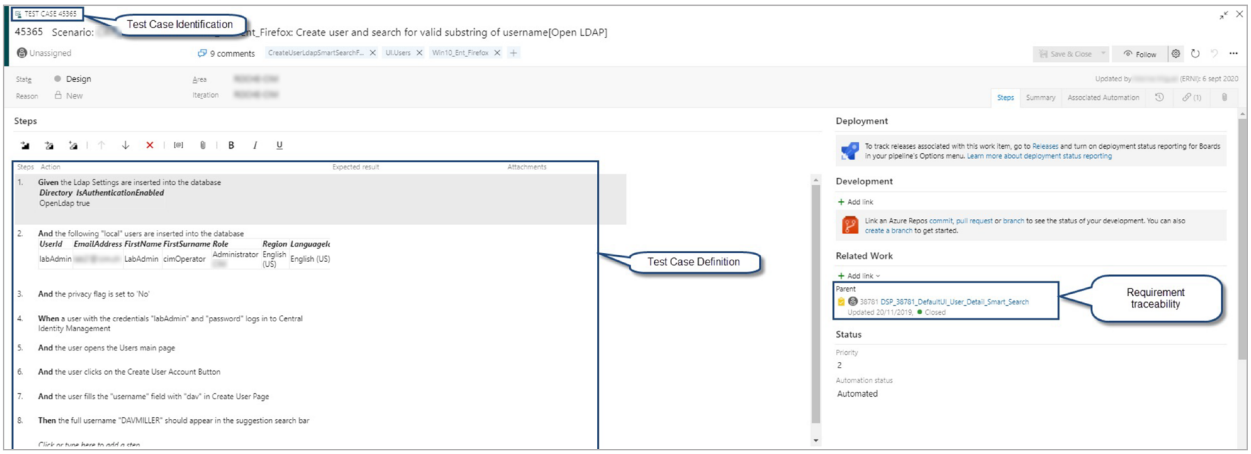


5. Verification *and Validation*

In general, verification stands for the process of checking whether the developed product it complies with the specified requirements, whereas validation checks if the intended use has been met and thus usability specifics are fulfilled as well. When it comes to **medical device software development, the design verification checks if the device was designed right. On the other hand, the design validation process answers the question or whether the right device has been designed.**



The main goal of the verification process is to confirm that the **Design Output** meets the **Design Inputs;** therefore, a **Design Verification plan** will be developed which will define how it is going to be ensured that each requirement (all level requirements) is checked. In this point, the process automation plays a key role in boosting the efficiency of the development; therefore, is highly recommended to use automated testing – as well as, automated traceability – to connect the requirements, the developed code and the tests.



The **validation process** is not part of the software development process, as the scope it covers is the complete medical device product.

The main goal of the validation process is to check whether the right device has been designed, and it is proved by using objective evidence which shows that the medical device meets the user needs and the intended use. It can include certain aspects such as initial production units, software validation, usability validation, clinical evaluation and the use under specific environmental conditions. As a summary, all generated documentation, such as test plan, test cases, test execution records, and test results, should be maintained as a part of the design records. **Validation, in its entirety, is not the result of a single activity, but the collection of results from all validation activities.**

With the aim of adding clarity, figures and pictures of medical device software design are attached. It is relevant to specify that the usage of certain tools are just examples of the framework implementation. However, similar tools can be used as long as they are validated for it.

Regulatory *add-on*

From SW developer to MDSW manufacturer: When MDSW developers decide to bring their own products to the market themselves, they are actually deciding to become a legal MDSW manufacturer, and this new role has legal implications they need to understand fully in order to successfully pass through the whole certification process and adequately fulfil the post-certification obligations.

6. Our *experts*

Dominique Monferrer
Medical Device Director
dmonferrer@asphalion.com

Albert Alsina
Medtech Industry Leader
albert.alsina@betterask.erni

Talyta Carteano
Medical Device Manager
tcarteano@asphalion.com

7. Contact *us*

ERNI

Spain | Switzerland | Germany | US | Singapore | Philippines

Contact: knowledgehub@betterask.erni

ASPHALION

Spain | Germany | UK |

Contact: info@asphalion.com