TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Department of Mechanical and Industrial Engineering

SMART MEDICAL DEVICE DEVELOPMENT FOR CLASS IIA MEDICAL DEVICE IN A COMPANY HEALTHCODE AI OÜ

NUTIKA MEDITSIINISEADME TOOTEARENDUS KLASS IIA MEDITSIINISEADME NÄITEL FIRMAS HEALTHCODE AI OÜ

MASTER THESIS

Student: Mihkel Siinmaa

Student code: 178159

Supervisor: Tauno Otto, Head of Development

Tallinn, 2019

AUTHOR'S DECLARATION

Hereby I declare, that I have written this thesis independently.
No academic degree has been applied for based on this material. All works, major viewpoints and data of the other authors used in this thesis have been referenced.

"....." ..... 201.....

Author: .....
/signature /

Thesis is in accordance with terms and requirements

"....." ..... 201....

Supervisor: .....
/signature/

Accepted for defence

"....."..... 201... .

Chairman of theses defence commission: .....
/name and signature/

Department of Mechanical and Industrial Engineering

THESIS TASK

Student: Mihkel Siinmaa 178159

Study programme: MARM, Industrial Engineering and Management

Main speciality: Industrial Engineering and Management

Supervisor: Head of Development, Tauno Otto, +37253090118

Thesis topic:

With the new Medical Device Regulations (MDR) in the European Union, many products on the market are going to get re-classified from Class I Medical Device to Class IIa Medical Device. This thesis observes the requirements for Class IIa Medical Device and offers a process for companies to follow when starting to build their Medical Device products. An example of project management on behalf of HealthCode AI OÜ is given and how the company applied requirements of Medical Device Regulations to develop their product.

Euroopa Liidu poolt sätestatud uued Meditsiiniseadme Regulatsioonid muudavad paljude meditsiiniseadmete klassifikatsiooni Klass I meditsiiniseadmelt Klass II meditsiiniseadmeks. Antud uurimustöös vaadeldakse Klass II meditsiiniseadmele sätestatud nõudeid ning pakutakse firmadele protsess, mida jälgida, kui hakatakse meditsiiniseadmeid valmistama. Projektijuhtimine on kajastatud firma HealthCode AI OÜ näitel, põhinedes meditsiiniseadme arendusele selles ettevõttes.

Thesis main objectives:

1. Analysing MDR and creating processes for developing Class IIa medical device products.
2. Applying MDR requirements to HealthCode AI OÜ smart device development.
3. Evaluating HealthCode AI's proposed patient management flow.

Thesis tasks and time schedule:

No	Task description	Deadline
1.	Description of Medical Device	05.03.2019
2.	Description of Software as a Medical Device	05.03.2019
3.	Description of software development methods	20.03.2019
4.	Medical device Class IIa requirements	20.03.2019

5.	Choosing the suitable development method for HealthCode AI	31.03.2019
6.	Example case of showing decision making process in HealthCode AI	31.03.2019
7.	Analysing Medical Device Quality Management System	10.04.2019
8	Analysing Medical Device Product Life Cycle requirements	20.04.2019
9.	Creating a process for developing a Medical Device	28.04.2019
10.	Creating two process models to evaluate the efficiency of proposed HealthCode AI patient management flow	12.05.2019
11.	Analysing model results	12.05.2019
12.	Analysing user feedback and usability of the software	18.05.2018
13.	Conclusion	26.05.2018

Language: English **Deadline for submission of thesis:** "....."....201….a

Student: …….. ..... "....."....201….a

/signature/

Supervisor: …… ..... "....."....201….a

CONTENTS

PREFACE

This thesis topic was initiated by my everyday work in HealthCode AI OÜ. The classification of Medical Device has been re-defined by the European Union in the new Medical Device Regulations. These changes are going to affect companies who previously categorized under Class I medical device the most, especially software companies such as HealthCode AI, who are developing software that is using sophisticated algorithms, machine learning or artificial intelligence. Many of these companies' devices will categorize under the new Medical Device Regulations to Class II device. This means that companies need to carry out medical evaluation of their devices. This thesis follows the new regulations and creates guidelines for software companies to develop their devices in order to achieve Class II certification marking. Implementing these new requirements into project management is shown by the example of HealthCode AI and their software development. The effects of the software are evaluated by using Arena Simulations to construct models for patient management flow. These models are constructed with the help of four Estonian physicians who gave input data that was necessary in building the simulation. Simulation results are analysed and the cost efficiency to Estonian central payer is analysed.

I would like to thank my family for supporting me in writing this thesis, HealthCode AI for giving me the opportunity to work with them and Tauno Otto for being my supervisor and guiding me through the process.

Keywords: medical device software, software development life cycle, management, Medical Device Regulations, Medical device class IIa.

List of abbreviations and symbols

AI – artificial intelligence

EHR – electronic health record

EU – European Union

EC – European Commission

MD – medical device

MDD – Medical Device Directive

MDR – Medical Device Regulation

ME – medical evaluation

ML – machine learning

MVP – minimum viable product

NoBo – Notified Body

RAD – rapid application development

QMS – Quality Management System

SaMD – Software as Medical Device

U.S. – The United States

INTRODUCTION

The classification of Medical Device has been re-defined by the European Union in the new Medical Device Regulations 2017/745 (MDR), effective from 2 April 2017 [1]. New requirements for medical device as a software will go into force in 2020 which will lead to re-classification. Until now, the majority of medical device software was classified under Class I, but the new regulations state that whenever software interferes with diagnostics or treatment, it will be classified as Class IIa medical device. Therefore, most Class I medical devices will be classified as Class II medical device. The importance of this change is that Class II medical devices require medical evaluation and implementation of respective quality management system. This adds significant workload to production and on top – if done incorrectly the company will not be able to obtain CE marking which is necessary for getting their product on the market. Costs for medical evaluation are significant, starting from hundreds of thousands of euros, and the time to perform evaluation might be up to couple of years. All this means that planning and execution of the project has almost no space for errors.

This thesis analyses the requirements for Class IIa medical devices and creates a process of steps that need to be followed in order to obtain the CE marking. A project management example given shows how company HealthCode AI made decisions in order to develop their product in accordance with the MDR.

Before deciding the medical evaluation process, it is critical to pre-assess the estimated outcome. Meaning that due to a lengthy and costly evaluation process it is crucial to perform close to life modelling of potential outcomes, to finetune the development and evaluation process. The final part of this thesis will do that: we evaluate HealthCode AI's proposed model of changes to primary care patient management flow and the cost-efficiency effect this change will have. The model has been constructed in Arena Simulation for academic purposes. Values for the model were gathered from four Estonian physicians. As software plays an important role in that model the usability was evaluated based on feedback gathered from user testing. User testing was conducted in face to face settings where every user was given the same task. User activities were recorded by using default recording software provided by the computer. User testing was conducted with 8 users between the ages of 18–45. The recordings were analysed by using usability key performance indexes and calculated in Microsoft Excel. The introduction chapter gives an overview of medical device as a software and the need for big data analytic software in healthcare sector.

1 OVERVIEW OF SMART DEVICES IN HEALTHCARE INDUSTRY

The European Union in Regulation 2017/745 defines “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. [1].

1.1 Medical Device classification

Medical Devices in Europe have been classified into 4 classes. The idea behind classification is the intent of use of the device and the inherent risk of that device to the patient or users. Devices are classified into a 3-tiered system ranging from Class I to Class III with the risk level as follows: Class I (including Is & Im) – lowest risk, Class II(a,b) – intermediate risk, Class III – highest risk [2]. All Class Is, Im, IIa, IIb and III medical devices require the intervention of third party: the so-called Notified Body. The classification rules are set out in Annex IX of the directives. Classification of MD have been brought out by 22 Rules which help to define device class. The main subject of these 22 Rules are as follows [1]:

- Non-invasive device
- Invasive device
- Active device
- Specific rules

1.2 Software as a medical device

As the need for big data analysis in healthcare industry grows, the need for big data analytic software grows. This need has created specific software that is used in healthcare. Software as a medical device is software that helps healthcare providers make decisions or diagnose their patients. MDR defines software as a medical device in Rule 11 under Annex VIII [1].

This definition determines that every software that helps to make decisions about the patient's health will be categorized under Class IIa. For example, this could mean that every application on mobile phone that gives users feedback on their health, diet, vital signs (such as heartbeat, pulse), could be categorized as a Class IIa software which requires medical evaluation. To put it into perspective, let's say that there is a software that asks the user three questions: what type of alcohol they drink, how often and how much. The information provided by this software could be used for marketing purposes; in this case – as marketing software – it doesn't have any requirements. But if we forward this information to a doctor and this information allows the doctor to make decision about the patient's treatment, then this information will be categorized under Class IIa MD and it will need to have a medical evaluation conducted.

1.2.1 Requirements for software as a medical device

Depending on the class of the MD software, certain requirements are set for the software to secure the safety of the patient. For software class IIa these requirements are shown in Figure 1.1. If all these requirements are met and the ME has been successful companies will be given a CE marking by an institution authorized by EU and the software will receive a UDI – Unique Device Identifier. The MDR [1] says that a new UDI-ID shall be required whenever a modification is made that changes interpretation of data – the change of algorithms, database structure, architecture or operating platform. Could this mean that companies using Artificial Intelligence or Machine Learning need to revaluate their software every time their algorithms change due to increased data sets? This is something that remains to be seen, as there are no sample cases available to be evaluated.

1.2.2 MDR conformity assessment procedure for MD Class IIa

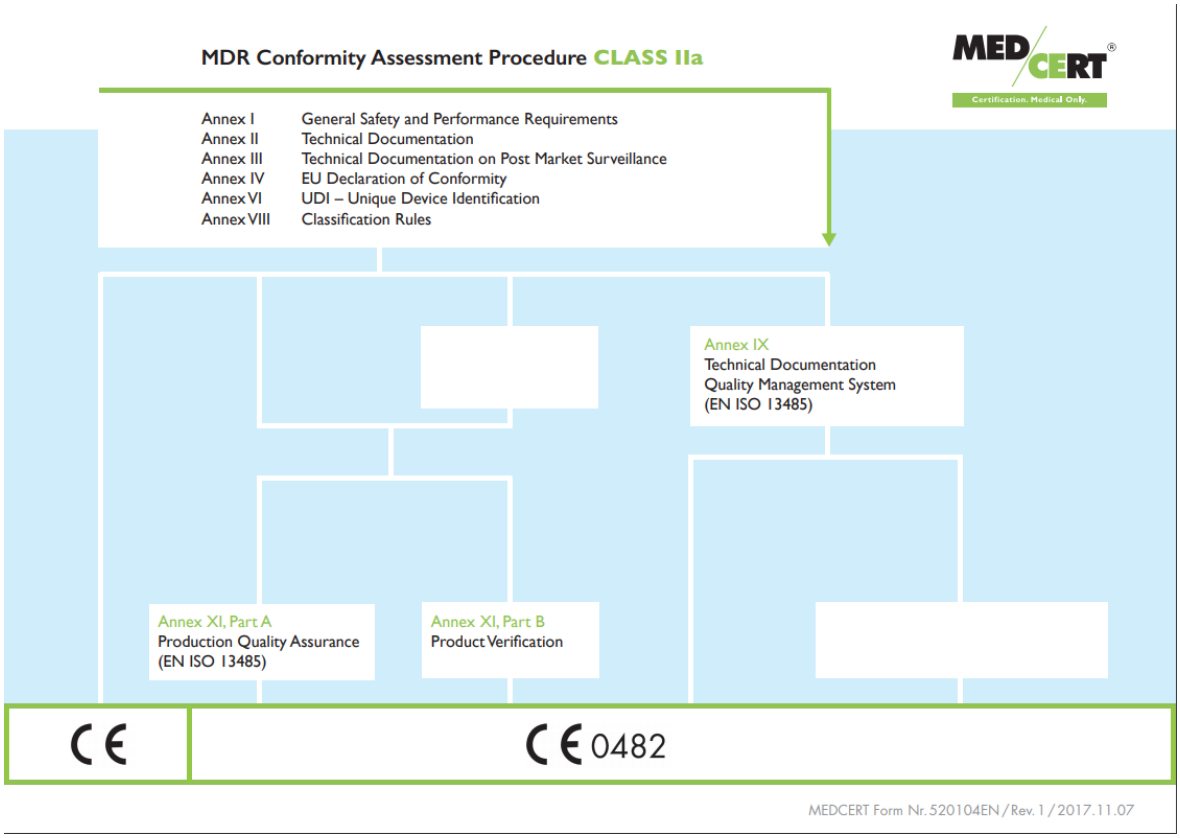


Figure 1.1 MDR conformity assessment procedure [3]

Figure 1.1 shows the criteria that the MD Class IIa must uphold and follow. Important annexes for software development in MDR are Annex IX and Annex XI. These give an overview of documentation that is required for companies to produce. These must be produced in order for the notified body to accept their device as a MD. Companies are required to show that they have technical documentation, an established quality management system, that production quality assurance in quality management system is being followed and that product verification has been done.

1.3 Innovation in healthcare industry

The healthcare industry has gone through huge digitalization like many other industries and started using Electronic Healthcare Records and Healthcare Information Systems [4]. This has generated huge databases in digital form which now require analytic tools to be analysed. Although the healthcare industry is one of the most innovative sectors it has fallen behind lately in innovation and technology adoption. The reasons are [4]:

- the unavailability of appropriate IT infrastructure,
- huge investment costs associated with implementing analytical tools,
- data privacy & security issues,
- fragmented data ownership and technical challenges such as data quality & multi-dimensionality of data.

One study identified lack of evidence of practical benefits as a major cause behind the reluctance for using Big Data analytics in healthcare [4].

Shared decision-making approach has become more and more popular amongst physicians. This method engages the communication between physicians, patients and potential others [6]. Most of the time decision making is based on personal experience, the experience from colleagues and data from verified sources like clinical studies or specific devices like X-ray or MRI machine. Physicians do not trust new technologies that do not have validated clinical proof. Development of the technologies is expensive, and no one wants to pay for the new technologies – often reimbursement from the government side is required. Newly developed products are often not user friendly and slow down the physician's workflow instead of making it faster. Many physicians see technology as impersonal [7]. These reasons have led to the stagnation of technology innovation in patient management in primary care setting.

1.3.1 The need for Big Data analysis in Healthcare

Reports say that data from the United States healthcare system alone reached 150 exabytes in 2011. At this rate of growth, big data for U.S. healthcare will soon reach the zettabyte (1021 gigabytes) scale and, not long after, the yottabyte (1024 zettabytes). Kaiser Permanente, the California-based health network which has more than 9 million members, is believed to have between 26.5 and 44 petabytes of data from EHRs, including images and annotations [5].

This amount of data can only be analysed with the help of big data analytic tools. Depending on the intent of use for these tools, they could be categorized under Software as a Medical Device. Although in order to use of big data analysis in healthcare setting, it is necessary to follow the regulations set forth in MDR, the end outcome has tremendous benefits.

By digitizing, combining and effectively using big data, healthcare organizations ranging from single-physician offices to large hospital stand to gain significant benefits. Potential benefits include detecting diseases at earlier stages when they can be treated more easily and effectively; managing specific individual health problems as well as larger population health issues, and detecting health care fraud more quickly and efficiently [5].

It is estimated that big data analytic software can enable more than $300 billion in cost savings per year in U.S. healthcare [5].

Table 1. Data Market by Industry, 2016-2017-2018-2020 (€, Million)

Industry	2016	2017	2018	2020
Healthcare	1,846	2,031	2,221	2,389
Information & communication	5,865	6,531	7,248	7,979
Financial services	11,816	13,145	14,521	15,860
Professional services	8,490	9,117	10,118	11,229
Total EU28	**59,496**	**65,286**	**71,593**	**77,769**

Source: European Data Market Monitoring Tool, IDC 2019

1.3.2 Using AI and ML to analyse Big Data

AI and ML platforms are capable of analysing huge amounts of data. There are two main factors that allow these platforms to work correctly. Algorithms behind those platforms must be correct and the data feed into the platforms must be quality data. This means that the machine will learn on the data it receives. If the data is wrong the machine will make wrong conclusions and if the algorithms are incorrect the quality of data does not matter – the result will be incorrect. This is a paradox which can only be controlled by experts once the platforms start producing results.

The aim for these platforms in healthcare is to support physicians and nurses in decision making and diagnosing the patient. Traditionally, a healthcare professional gets their data form the workplace and gathers experience over many years, throughout their career (from individual work, consultation and collaboration with colleagues, and continuous self-improvement such as keeping up with latest clinical literature and studies, participation in conferences, etc). In comparison, an AI and/or ML platform does not need decades to master a complicated topic – it is able to acquire information from big data sets, analyse them, and learn the necessary information in a relatively short amount of time.

Big data analytic tools can help to identify the probability of events. This could be used to determine what illness or condition a patient might have by using the tools to analyse pre-existing conditions, symptoms, events and comparing them to known results. Pattern recognition technique is used to analyse X-ray images which help to find pre-determined signs and give feedback to doctors once that sign is detected. Sometimes these signs are very specific to a problem and a doctor might see it only once in a lifetime. Using analytic tools, the machine can gather data from many centres and get 10 times more knowledge about cases that might occur only 1:100 000 patients. This allows the platforms to give feedback to doctors and advise them to evaluate that possible hypothesis.

2 HEALTHCODE AI SOFTWARE DEVELOPMENT – LEIA

This chapter follows the project management in developing software as a medical device in the company HealthCode AI OÜ. Description of the company and overview of their device in development is given. The project management focuses on one step in the chain of development and explains which methods and steps were chosen. Regulations which affect the decision making of project management are brought out and their effect on the decisions that where set by the project manager are described. Different methods of software development and MDR requirements are analysed and a method is developed that is suitable for HealthCode AI in developing their device. The usage of this method is shown in a case example that describes developing part of the product and getting feedback and reworking when needed.

2.1 HealthCode AI background

HealthCode AI OÜ is a software development company focusing on the healthcare industry. Currently they are developing software that allows to improve a physician's workflow efficiency by 22%. This is achieved by using machine learning methodologies and artificial intelligence to analyse big data for the physician. The device they are developing is classified as Class IIa software. This product is offering a new way to manage the flow of patients and is shown in Appendix 8.

The idea of developing this software comes from studies have shown that presenting patient data to the physician before the consultation is going to make the consultation 70% more effective for regular patients and 40% more effective for chronical disease patients [28]. To gather information HealthCode AI has developed an AI that has the capability to question patients about their symptoms. This software gathers information from patients by using a machine learning platform which imitates a physician's diagnostic thinking. This data is then analysed and presented to the physician in a visualized manner. HealthCode AI has developed a unique disease visualization method that allows physicians to get an overview of the patient's symptoms, complaints and disease progression in a few seconds.

One of the problems with big data analytic tools for healthcare industry is that they are not user friendly and often take even more time away from physicians than they give back. Taking this into consideration, HealthCode AI is developing its products together with physicians and listening to their feedback throughout the product life-cycle.

Error! Reference source not found. in Appendix 8 shows the proposed patient management flow w hen starting to use HealthCode's software to do patient questioning. This flow is analysed in chapter 4 by using Arena Simulation to see the effect of the system and the possible financial benefits for central payers. The figure above describes two patient management flows. The first one shows how patients are currently managed and how much time it takes in each step and the second one provides the potential patient management flow when using software to carry out patient questioning.

2.2 Software development project management

2.2.1 Project overview

The objective of this project is to develop minimum viable product (MVP). This device must be able to gather information from the patient and deliver this information to the physician. The goal of the MVP is to be able to go into medical evaluation and start the process of achieving Class IIa medical device certification. ME is intended to be carried out in primary care setting in Estonia.

In this project there are two main parts of the software that need to be completed, these are:

- Patient Portal
- Physician Portal

The timeframe for the product development is:

01.11.2018 – 30.04.2019

Required human resource to finish the MVP is:

- Knowledgebase developer
- Project manager
- Developer
- User experience / user interface designer

Functionalities that are required for this product phase to be met by the end of the deadline are:

- Machine learning platform
- Patient questioning
- Disease visualization
- Diagnose hypothesis creation
- Decision making support

The project is regarded successful when the patient is able to describe their health problems through HealthCode's software and send it to their physician. This is evaluated in chapter 4 under usability testing. The physician must be able to receive the patient's information and see it in a visualized manner. This must be achieved by 30.04.2019.

2.2.2 Medical device project management specifics

Developing a product that is classified as an MD by MDR requires the establishing of additional rules that must be followed during the development process. These rules are described in more detail in chapter 3. For project managers it is important to understand under which MD class does your product classify as, and what are the requirements for that certain class. As mentioned in paragraph 1.2.2, project managers must take into consideration several requirements set by MDR.

For understanding MD development process, a flow chart of actions was created (Figure 2.1). This flow chart was developed to give the team an overview of the regulatory steps that are required for developing SaMD.

Project managers must take into consideration all off these steps before going into the development. For HealthCode AI these steps were followed when creating the software development project plan.

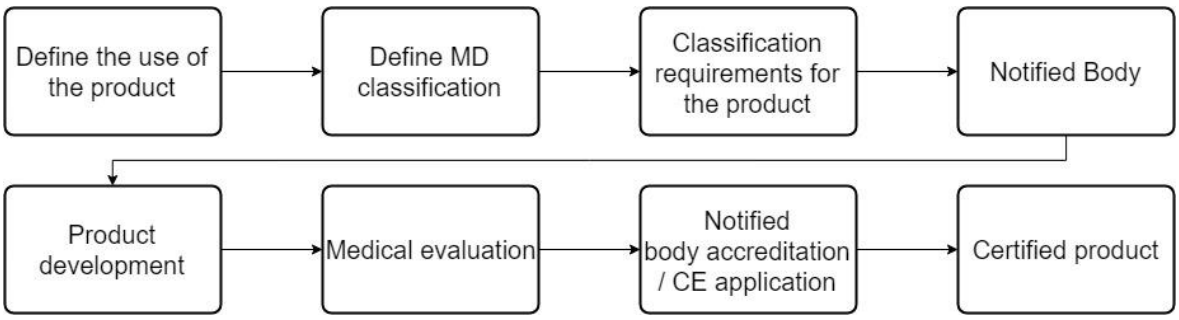


Figure 2.1. MD development process steps

Use of the product:

The first step is to describe the use of the product. This will define if the product goes under MD. After the intent of use of the product is identified, the company must check if their description fits with the description provided by MDR. If it does, the device classifies as a MD.

MD classification:

Once the product is classified as an MD it is necessary to define the specific classification of the product. The classification of the product has been described in detail in section 1.1.

Classification requirements:

Each MD classification has different requirements. In this step the company should make sure which requirements they need to meet and plan those into their everyday workflow. This step gives strict guidelines which must be met during the development process of the product.

Notified Body:

Notified body is a third party organization that offers conformity assessment to products before they are placed on the market. The NoBo helps companies to understand which requirements are necessary in order to achieve certification.

Product development:

In this part it is necessary for the project manager to know which documentation must be developed together during the device development process. Guidelines set in MDR must be followed in order to receive the certification marking.

Medical Evaluation:

ME is required for all MD products that are classified as Class II or higher. The aim of a ME is to show that the product being developed is safe and does all the functionalities that were defined in the use of product.

Accreditation and applying for CE marking:

Accreditation is done by NoBo to evaluate the result of the ME. Once the evaluation is completed the NoBo will inform the CE authority of their decision. In Estonia the CE authority is Terviseamet (Health Board).

Certified product:

The certification for the product will be given out by the local authority in the country (designated by the European Commission).

2.3 Software development requirements for Class IIa

Only software that is used as an MD must have a CE marking. For other software this is not required. This raises additional requirements that project managers must include in their project planning. These requirements are specified in chapters below in order for companies to achieve CE marking for their Class IIa software. It is important that these processes are followed throughout the device development lifecycle.

2.3.1 Standards required for medical device Class IIa development

There are three main standards that are highlighted in the MDR [1] when starting to develop MD. It is necessary for companies to follow these standards to ensure that development process for the device is done correctly and once the device is on the market it complies with the rules.

The following standards are required to be adapted in order to achieve the certification for software:
IEC 62304:2006 – Medical device software – Software life cycle processes
ISO 13485:2016 – Medical device – Quality management system – Requirements for regulatory purposes
ISO 14971:2007 – Medical device – Application of risk management to medical devices

Most relevant standard for software developers is IEC 62304:2006 – this gives the guidelines for the software development and its life-cycle processes. For project managers this standard shows which additional tasks developers must do in order to reach compliance. This standard provides requirements for each life-cycle process. Each life-cycle process is further divided into a set of activities, with most activities further divided into a set of tasks [8]. Although this standard focuses more on the development process, it is important that the development follows the quality management and risk management system set in the International Standard ISO 13485 [9] and ISO 14971.

ISO 13485

Project managers must use this standard to set up the processes inside the company. The aim is to construct a quality management system for your company which allows to achieve the CE marking for the device being developed. It is based on a process approach to quality management. Any activity that receives input and converts it to output can be considered as a process. Often the output from one process directly forms the input to the next process.

For an organization to function effectively, it needs to identify and manage numerous linked processes. The application of a system of processes within an organization, together with the identification and interactions of these processes, and their management to produce the desired outcome, can be referred to as the “process approach.”

When used within a quality management system, such an approach emphasizes the importance of:
a) understanding and meeting requirements;
b) considering processes in terms of added value;
c) obtaining results of process performance and effectiveness;
d) improving processes based on objective measurement.

2.4 Software development methodology

This chapter will give an overview of different software development methodologies and explain why HealthCode AI chose one of those. Pros and cons of different methods are evaluated and an evaluation matrix is used to choose the most suitable method for HealthCode AI. Overview will be given of the development method that HealthCode AI is using and how this is implemented by an example case.

2.4.1 Overview of software development methods

Waterfall development methodology

Waterfall method consists of phases. Each phase consists of predetermined tasks and once these tasks are 100% complete, development is allowed to move on to the next phase. Once one phase has been completed it is locked, this means that there is no going back to previous phase to implement changes. The requirements for each phase must be clear before going forward with it. The testing and documentation takes place at the end of each phase – this helps to maintain the quality of the project [10].

Pros

- Requirements are set in stone before development starts, which is convenient for developers
- Easy to implement and understand
- After each phase proper documentation is produced

Cons

- If the client wants requirement changes, these will be not implemented during current development process
- Phases are locked after development and if problems arise further down the development they will not be attended to.
- Value delivery at the end of project, which creates high risk of client expectations not being met.

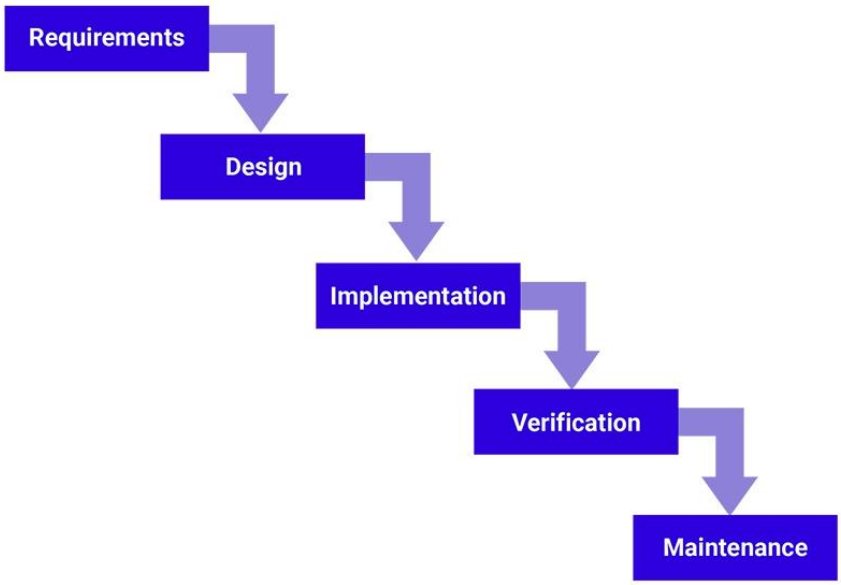


Figure 2.2. Waterfall model, [27]

Rapid application development method

RAD is a complete approach to information systems development as it covers the entire life cycle from the idea to the delivery [11]. The key objectives of this method are:

- High quality systems
- Fast development and delivery
- Low costs

This method has four phases which are shown in Figure 2.3. User design and construction are in a continuous loop until the customer confirms that all requirements are met.

Pros:

- If the requirements are well defined the development is fast
- Good for projects which are small and time sensitive

Cons:

- Requires highly skilled developers

- Necessary for users to know what they want

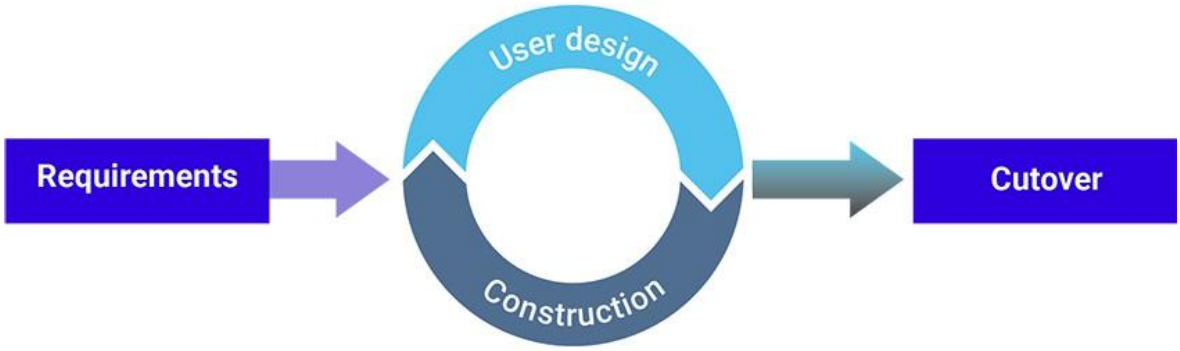


Figure 2.3. Rapid application development method. [27]

Agile development methodology

Agile method comes from the agile manifesto which outlines a set of four values [12]:

"*Individuals and interactions over processes and tools. Working software over comprehensive documentation. Customer collaboration over contract negotiation. Responding to change over following a plan*."

In addition to its 4 values, The Agile Manifesto also outlines 12 principles for agile development practices. These 12 principles bring the companies attention to early and continuous delivery of software and constant emphasize on continuous attention to technical excellence. They describe a culture in which change is welcomed, and the customer satisfaction is the main priority. The aim of Agile is to align software development with business goals.

Pros:

- Minimize risk by constant feedback loop
- Tight communication between different teams in the company
- Software development in small iterations
- Good for finding defects and expectation mismatches early on

Cons:

- Dependant on good communication between business and IT sides
- Time commitment from users required
- Developers must complete each feature within each iteration

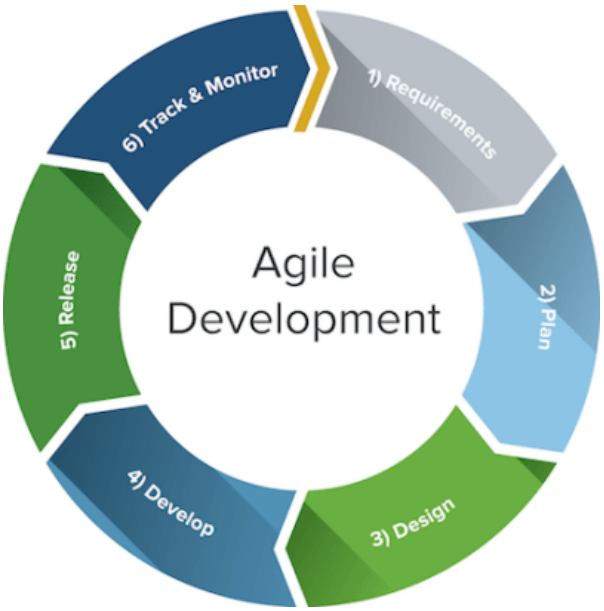


Figure 2.4 Agile Development method

DevOps deployment methodology

DevOps is often viewed as a methodology that focuses on communication, collaboration and integration between software developers and IT operations. It is comprised of four aspects: culture, automation, measurement and sharing. One of the main goals of DevOps is to tackle the problem of having development team and operations team not communicating, collaborating with each other. DevOps aim is to launch products, iterations, fixed as soon as possible with a smooth pipeline [13]. Applying DevOps allows companies to produce a so-called pipeline to that allows launch iterations faster as the system aims to automate version releases.

Pros:

- Faster time to market, as the software release is done in a pipeline
- Small releases mean lower rate of failures on release
- Automated continuous deployment

Cons:

- Different environments used by different departments can allow undetected errors go through to live version
- Some industries have regulations that require extensive testing before a project can move to the operations phase

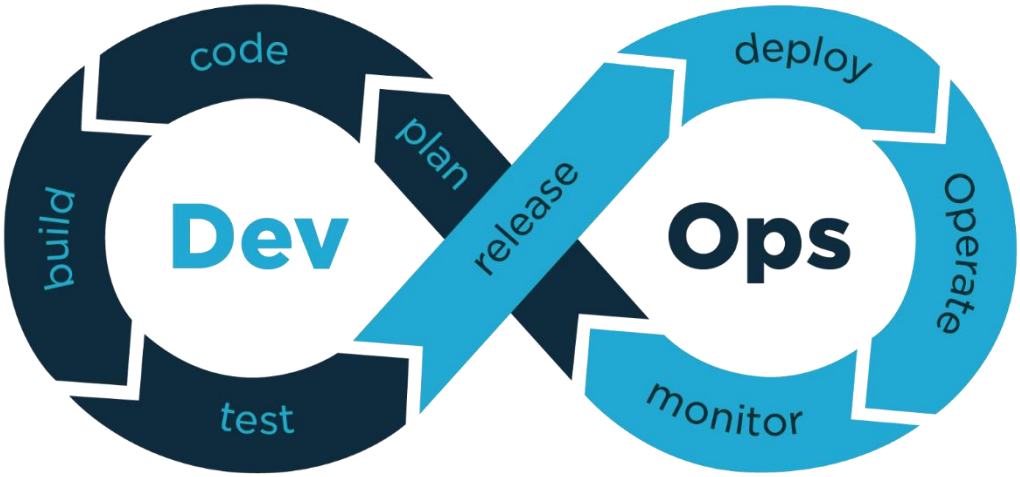


Figure 2.5. DevOps model

2.4.2 HealthCode AI software development methodology

HealthCode chose the software development method by evaluating the pros and cons of most common development methodologies, team experience with the usage of different methodologies, and MD specific requirements for Class IIa.

For decision making an evaluation matrix was created where each team member had to evaluate each topic on scale 1-5, with 1 representing low suitability and 5 representing high suitability. The sums of each column were added together and final decision was made by the highest value of each column

Table 2. Development method evaluation matrix.

	Waterfall				RAD				Agile development				DevOps			
Customer satisfaction	2	2	2	2	2	2	2	2	4	4	4	4	3	3	4	4
Speed	1	1	1	1	4	4	4	4	3	4	5	3	2	3	4	3
Familiarity to method	1	1	1	1	1	1	1	1	4	4	2	2	2	2	1	1
Iterations speed	1	1	1	1	3	3	4	4	4	4	4	4	4	4	4	4
UI/UX implementation	1	1	1	1	3	3	3	3	4	4	4	4	3	3	3	3
Future Scalability	1	1	1	1	2	3	2	2	3	4	4	4	5	5	5	5
Technical documentation	4	4	4	4	2	2	2	2	5	4	4	4	4	5	3	3
Implementing with QMS	4	4	4	4	3	4	3	2	4	5	4	3	3	4	5	5
Total	60				83				123				112			

Based on the result of the evaluation matrix the Agile development methodology was chosen. The next chapter will describe how the company started using this method in their development process.

2.4.3 Agile method and QMS for medical device development

Agile in its core is meant to be adaptable to the context in which it is being applied. This idea and the principles that Agile development stands for were the main reason for adapting this method for HealthCode. The idea that combining quality management main principle – customer satisfaction first – and the flexibility of Agile development led to the creation of HealthCode development methodology.

As for adaptability, these main principles where taken from the Agile manifesto and added to the day-to-day development process:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. Welcome changing requirements, even late in development.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business side and developers must work together daily throughout the project.
- Simplicity – the art of maximizing the amount of work not done – is essential.

HealthCode AI development methodology is shown in Figure 2.6. The main goal of this methodology is to learn what customers require, to bring these requirements to life by dividing them into small work packages and getting constant feedback on every function that is developed. Quick iterations allow to try out the requirements set by the customer, to see if they work or not and to keep the software constantly updated and built the way the customer wants it to be. When deciding which part to develop first, a meeting is held. The requirement that is being developed is rated by all the members from 1-5, with 1 meaning non-crucial and 5 the most crucial part. As each member gives their opinion to each function the sum is added together, and the priority list of functions is created. This method allows different members to evaluate and explain why they see some functions are more important or time consuming.

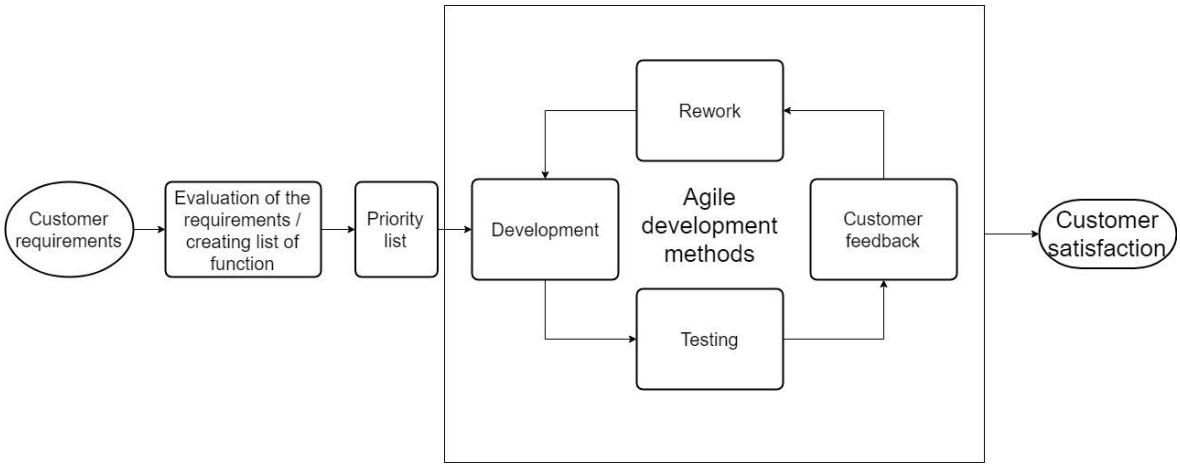


Figure 2.6. HealthCode AI development methodology

2.4.3.1 An example case on how priority list is created

Customer requirements:

Customer (physician) wants to see the name, sex and the age of the patient in the system.

Defining the requirements into functions:

The sex and the age can be received from personal identification number.

F1: Make registration personal identification number based

When registration is done ask the user to input their name

F2: Make personal information page

More functions that are required

F3: Save user data into database

F4: Let physicians choose between different patients

F5: Show patients information to physician

Creating a priority list by rating each function from 1-5.

Table 3. Priority list evaluation

Function	Member1	Member2	Member3	Total
F1	5	5	5	15
F2	2	2	1	5
F3	5	4	5	14
F4	1	1	1	3
F5	2	3	5	10

Final function development order as follows: F1, F3, F5, F2, F4.

2.5 Software development process

2.5.1 MDR Annex II: Technical Documentation

Annex II in MDR specifies the technical documentation required. This documentation describes the software design and development process and evidence of the validation of the software, as used in the finished device. This information shall typically include the summary results of all verification, validation and testing.

IEC 62304:2006

In paragraph 5.1.1 of IEC 62304, Software development guidelines are given as such: The manufacturer shall establish a software development plan for conducting the activities of the software development process appropriate to the scope, magnitude and software safety classifications of the system to be developed. The software development life cycle model shall either be fully defined or be referenced in the plan. The plan shall address the following [8]:

1) The process to be used in the development of the software system
2) The deliverables (includes documentation) of the activities and tasks
3) Traceability between system requirements, software requirements, software system test, and risk control measures implemented in the software
4) Software configuration and change management, including SOUP configuration items and software used to support development and;
5) Software problem resolution for handling problems detected in the software products, deliverables and activities at each stage of the life cycle

Figure 2.7 and Figure 2.8 give the overview of the software development flow in general. All the necessary steps in software development and maintenance are shown. These steps must be documented and the development methods, process must follow the company's quality management system.

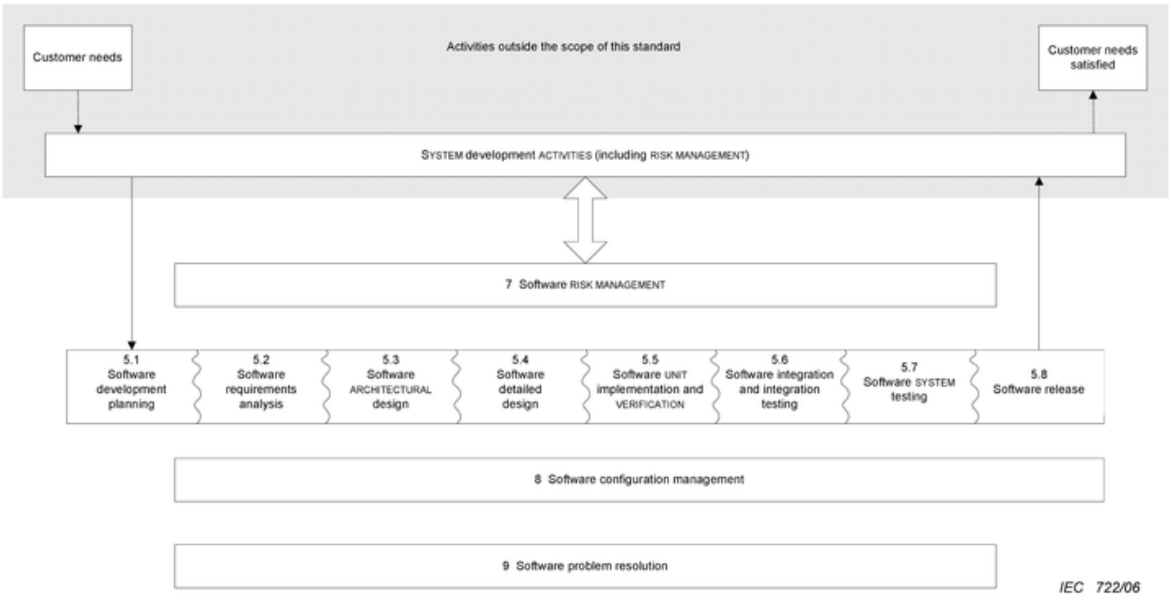


Figure 2.7. Overview of software development processes and activities

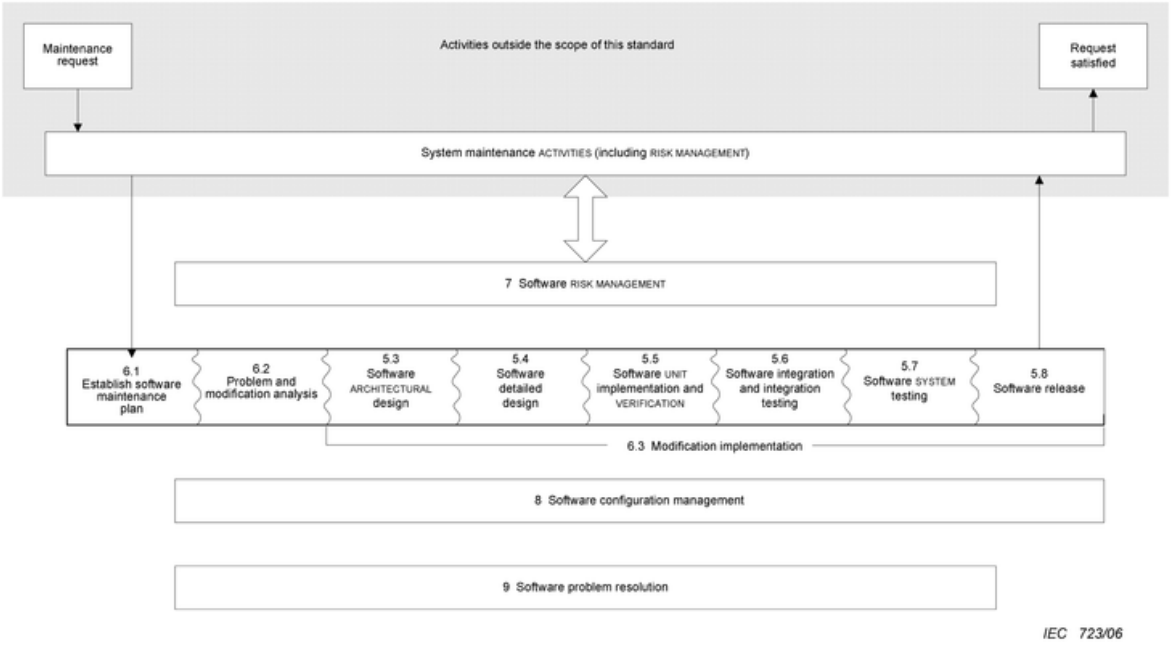


Figure 2.8. Overview of software maintenance PROCESSES and ACTIVITIES

Customer needs

As customer satisfaction is key element, HealthCode AI has developed its own methodology to get information from customers and to understand their needs better. This process is described in Figure 2.9.

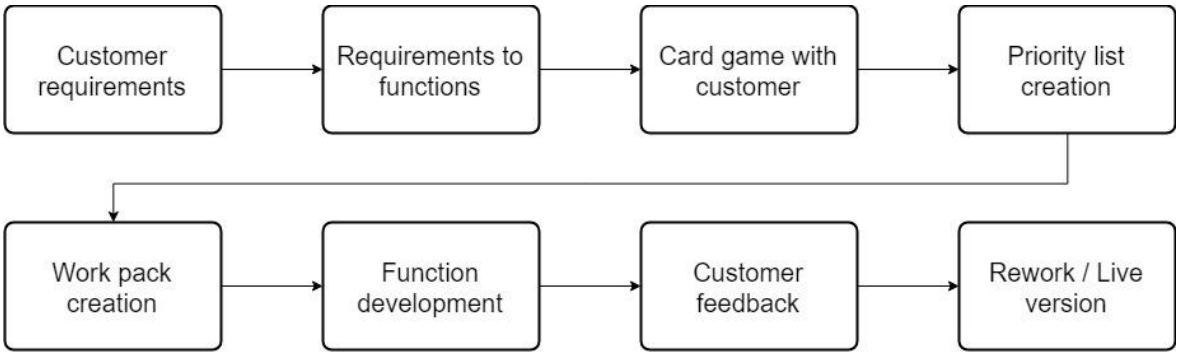


Figure 2.9 Customer need flow chart

2.5.2 Customer flow chart example use case

Customer requirements are gathered via an interview with customers. The aim is to understand the "pains and gains" of the user and find solutions together how to address them.

Information gained from interviews is then analysed and transformed into functions. For example, one of the customer pains that came out during an interview was that the customer doesn't often remember the names of their patients and once their patient enters the office they might have thinking about some other person entirely. This pain was developed into following functions:

1. Name
2. Sex
3. Age
4. User profile picture

When the functions are defined, a card sorting game is done with the customer. Card sorting is a method used to help design or evaluate the information architecture of a product. The UX designer asks users to group content and functionalities into open or closed categories. The result gives the UX designer input on content hierarchy, organization and flow. The end goal is to understand where information is laid out by the customer. In this case, the users placed the name, sex, age and picture information into top left corner.

Once it is defined how customers place the information, the priority list of functions is created. Priority list is created with the feedback from card sorting and decision-making matrix. For this particular case the most important information was Sex and Age, secondly Name and least important was user profile picture. Although the pain started with the customer not putting together the name and the face of their patient, the card sorting game showed that for our

customer the most valuable information is sex and age of their customer. When the priority list is created, the next step is to create work packages for each function so that it is possible to estimate the time spent on developing each function. Function development follows the logic shown in Figure 2.6.

When the development has finished, getting customer feedback is a must. This feedback is gathered by doing usability testing. Usability testing is the observation of users trying to carry out tasks with a product. Testing can be focused on a single process or be much more wide ranging.

When possible, HealthCode does A/B testing with their users. In A/B testing, alternative versions of a product are shown to different users and the results are compared in order to find out which one performs better. This testing gives insight which version works better and gives feedback on what customer like and dislike. After receiving feedback from A/B testing, either necessary rework is initiated or the function is sent to live version. In this use case the function was reworked due to customer feedback.

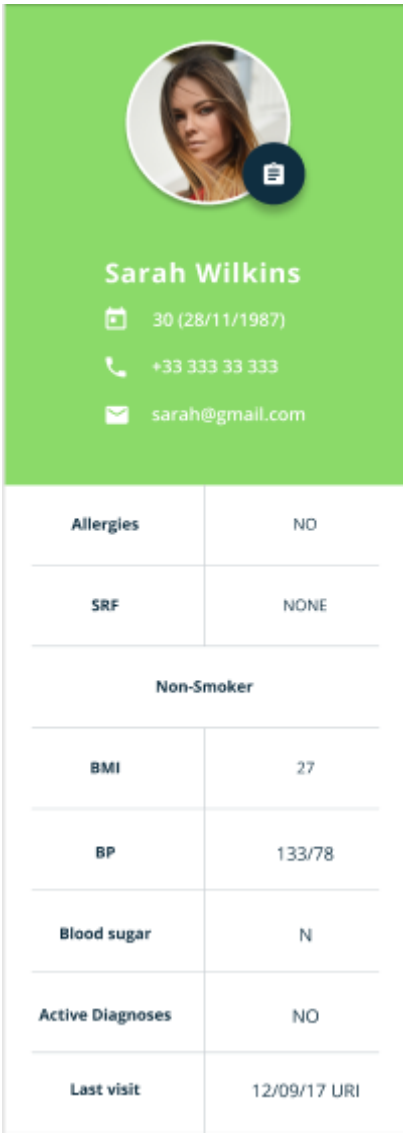


Figure 2.10 User test A

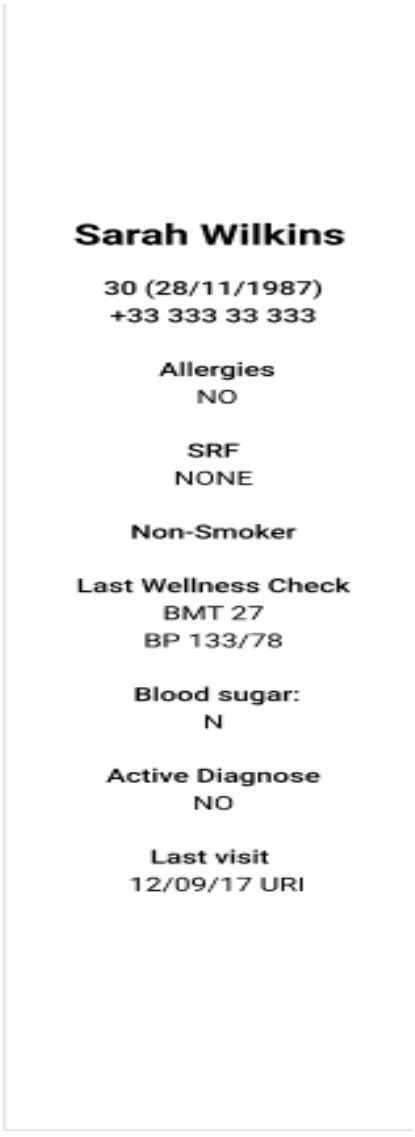


Figure 2.11 User test B

3 GUIDELINES FOR DEVELOPING MD CLASS IIA SOFTWARE

In order for a Class IIa software as medical device to be able to enter the market in the European Union, a CE certification is compulsory. This section identifies the steps necessary for obtaining the CE marking and the role of Notified Bodies (NoBos) in this process. Following these steps closely is of great importance, because if any issues are discovered in late stages, it may be necessary to start the process all over again. This is due to a requirement in the regulation, that says modifications to the software algorithm are not allowed without new certification. Therefore, it is crucial to develop the device following requirements and correct processes to avoid time consuming and costly re-work and re-certification process.

Both U.S. and EU have acknowledged that there is a disconnect between how quickly software can be iterated, and that cycle of approvals is lengthy. Authorities are interested to explore models under which medical device software might be released more quickly, but there are no effective solutions so far.

3.1 CE certificate process

The CE marking process flow was created and is mapped out in Appendix 5.

The first steps in the process are defining the product's Intent of use and finding out the classification of the medical device. It is important to note that any change in Intent of use means it is necessary to start the process all over again. After Intent of use and classification of the MD have been defined, the next recommended step is implementation of a quality management system (QMS).

3.1.1 Quality management system

QMS implementation and medical evaluation – not just for regulative purposes.

There is a wide concern that algorithms may mirror human biases in decision making. AI applications introduced in nonmedical fields have already been shown to make problematic decisions that reflect biases inherent in the data used to train them [29, 30]. Recently, a program designed to aid judges in sentencing by predicting an offender's risk of recidivism was shown to have a serious tendency for discrimination [29, 30]. Similarly, an algorithm designed to predict outcomes from genetic findings may be biased if there are no genetic studies in certain populations [29, 30].

The intent behind the design of AI also needs to be considered, because some devices can be programmed to perform in unethical ways. For example, Uber's algorithm tool 'Greyball' was designed to predict which passengers might be undercover law-enforcement officers, thereby allowing the company to identify and circumvent local laws [29, 30]. Also, Volkswagen's algorithm allowed vehicles to pass emissions tests by reducing their nitrogen oxide emissions when they were being tested [29, 30]. Analogously, private-sector designers who create AI algorithms for clinical use could be subject to similar temptations, programming AI systems to guide users toward clinical actions that would generate increased profits for their purchasers (such as by recommending drugs, tests, or medical devices in which they hold a stake or by altering referral patterns) but not necessarily reflect better care [29, 30].

It is clear that AI systems do more than process information and assist officials, therefore it is crucial to make sure that the AI based Medical device is acting according to the intent of the product and does not bring harm. Every developer of AI/ML based medical device carries significant responsibility to do everything possible to secure that. The minimum is to follow current legislation that defines the need to implement quality management system, and to perform medical evaluation.

Implementing quality management system

A QMS process flow is shown in Appendix 6. Additionally, Article 10 in MDR describes the minimum requirements of QMS for MD manufacturers [2].

The QMS sets up the company's culture and gives guidelines for the organization in dealing with quality of processes, procedures and devices, it shall define responsibilities, processes and management resources required in achieving compliance with the provisions of MDR. Medical Devices QMS must be developed by using ISO 13485 standard. This standard is a must requirement for a company to receive a CE marking. However, when developing SaMD, only relevant information presented in the standard must be taken into account. For example, packaging does not affect software development companies.

The ISO 13485 standard requires documentation of the processes, operations, and activities that make up the QMS. If the company decides to maintain low-detail documentation during an audit, they must show how this documentation is sufficient. Documentation is divided into two levels:

- Define how processes, operations, and activities shall be documented.

- Define which process outputs must serve as evidence and be maintained in the form of records of processes, operations, and activities

Understanding the different phases of the life-cycle of the MD is the basic stage in developing an effective QMS in the medical device industry. Basically, there are six major phases in the lifespan of a medical device from conception and development to disposal.

Figure 3.1 Six phases in the product life-cycle

In the premarket stage, it is important to ensure that the MD:

- Is developed according to customer, safety, and regulatory requirements
- Has been tested or clinically tried
- Performs as expected
- Is safe for use
- Complies with regulatory requirements

In the placing-on-market stage, it is ensured that:

- The vendor is registered.
- The MD is registered as required in each region where it is marketed.
- The performance and intended use of the MD are communicated correctly to the public.
- After-sales obligations like user support, complaint handling, or maintenance of user records are being pursued by the manufacturer or the vendor.

In the postmarket stage, it is ensured that:

- The use of the MD is being closely studied for relevant events that occur, like feedback from users, adverse events, or other new developments or changes in the area of the MD that require reaction.
- Systems for reporting and alerts are developed.
- The MD is adequately disposed of.
- The safety and performance of the MD that is in use are ensured and improved.

Process based QMS

Process based QMS starts by determining all key processes needed to finalize product. These are all the stages that must be done in order to finish the product. Once the key processes are defined the next step is to set them in logical sequences. In this stage the process workflow is defined. Once the workflow is set it is important to describe the necessary activities within a process. If regulatory requirement is applicable to a process more activities might be required. Also, all key processes that are needed to finalize product must be determined. If these steps are defined, then documentation to support processes must be created. Types of such documentation:

- Management-oriented process
- Process diagram/flowchart
- Documented procedure
- Work instruction
- Standard operating procedure

For each process a company must define inputs and outputs. Outputs must be identifiable and measurable. Assigning a responsible person to a process creates a relation between the organizational structure and the workflow.

Technical file

After the QMS has been implemented it is required for the company to produce a technical file. Minimum content of technical file should include [16]:

1) Table of contents
2) General information concerning the structure and use
3) European Commission declaration of conformity and classification 93/42/EEC
4) Name and address of the manufacturer
5) Product description
6) Product specifications
7) Product verification
8) Product validation

3.1.2 Notified Body

After QMS has been implemented, it is recommended for the company to choose a NoBo, although this can be done earlier as well, as NoBos can support companies during the development process.

Notified body (NoBo) is an organization designated by an EU country that carries out the conformity assessment procedure on medical devices. The reason for the existence of NoBos is that the European Commission does not have enough time or resources to evaluate each product themselves, so they have delegated this task to third parties (NoBos) who are regulated by local governmental bodies.

It is important that every product being brought into the EU market complies with the applicable legislation [14]. NoBo ensures that the MDs with higher risks are evaluated properly before being released onto the EU market. It is important to work together with the NoBo in order to make sure that the software development is done correctly and that it follows all the requirements set by MDR.

Notified body experience with ML / AI

NoBos have limited experience with machine learning / artificial intelligence solutions because earlier versions of medical regulations did not include specific rules regarding AI/ML based devices. There is also very limited guidance from regulations. Even more, MDR states that a recertification shall be required whenever there is a modification that includes new or modified algorithms. From machine learning angle, it means that company is required to recertify every time machine learning platform is trained, and a new algorithm is created. So it is clear that more advanced regulation is needed in terms of AI/ML devices to allow healthcare system to enjoy the full benefits of innovation.

Audit by Notified Body

For Class IIa medical devices, the company can choose a NoBo to audit them according to one of four defined conformity routes [15]:

1) an examination and testing of each product;
2) an audit of the production quality assurance system;
3) an audit of final inspection and testing; or
4) an audit of the full quality assurance system.

When developing software which uses AI, ML or specific algorithms it is suggested to avoid route 1 or 3. This is because every time the product's algorithm changes, it is required to undergo medical evaluation process once again. This requirement comes from MDR UDI chapter [2]. As every Class IIa MD needs a medical evaluation, which is a time and resource consuming process, routes 1 and 3 are not feasible for many companies. Companies with algorithm-based product should prefer the

route of quality assurance system (nr 4); with this option they need to prove that the device development process meets the requirements set by MDR and these requirements are followed.

3.1.3 Medical Evaluation

Next step is to conduct medical evaluation (ME) for the company's software. ME is necessary to provide evidence that the product can fulfil its purpose in the intended manner. The evaluation must be conducted in correct clinical settings, meaning that everything used in the ME must be documented and tested appropriately. For example, if physicians measure blood pressure, they must record the data from the measurement as well as the device they used for measuring.

Before going into medical evaluation, the company needs to [20]:

1) identify the Essential Principles that require support from relevant clinical data;
2) identify available clinical data relevant to the device and its intended use;
3) evaluate data in terms of its suitability for establishing the safety and performance of the device;
4) generate any clinical data needed to address outstanding issues;
5) bring all the clinical data together to reach conclusions about the clinical safety and performance of the device.

Results from these processes are documented in the medical evaluation report. Medical evaluation process is also described in Table 4.

Table 4. Medical evaluation process [17].

Medical Evaluation Steps		
Valid Clinical Association	**Analytical Validation**	**Clinical Validation**
Confirm that the output provided by your software matches its targeted clinical conditions [18].	Confirm that the software functions correctly, i.e. the software meets its specifications and data analyses are done to the appropriate level of accuracy.	Confirm that the use of the software's accurate, reliable and precise output data achieves the intended purpose in the target population. I.e. is the Intent of use being met.

Medical Evaluation results will be assessed by the NoBo that the company chose. In most cases, the NoBo needs real life evidence to make their decision. In most cases this means it is necessary to do a medical evaluation with real life clinical data or patients.

Finally, when the company has an established QMS, ISO 13485 standard certificate, and has completed the ME report, they are ready for the final CE marking audit. CE mark is given to a company without any time-restrictions. After CE mark has been obtained, the company's goal should be to maintain the ISO certification by regularly doing re-audits and re-testing of the product, because the regulators have the right to come and check the processes at any time.

3.2 Software development life cycle model

This chapter describes the necessary steps required to take while developing software as a medical device. This chapter will not take a look at the development methods nor architecture but will focus on the requirements set by regulations when developing Software as a Medical Device. These steps set in regulation are brought out in separate chapters and described more in detail in each chapter. A process flow was created for software development life cycle which is brought out in Appendix 7.

When starting to develop SaMD, companies must comply with the requirements set in the IEC 62304 standard. This standard provides the framework of software development life cycle processes.

Medical device software development starts with software safety assessment. Safety is divided into three classes which indicate the potential risk to the users the software might bring. The classification requirements and types are set in the IEC 62304 standard. Software classification is as follows [21]:
A: No injury or damage to health is possible
B: Nonserious injury is possible
C: Death or serious injury is possible.

When software has been classified the IEC 62304 standard gives indication which requirements are necessary to meet in every aspect of the software development life cycle. Each class has its own set of subclauses they need to meet in order to get certified. These requirements are shown in

Table 5. Software Documentation

Clauses and subclauses		Class A	Class B	Class C
Clause	**Subclauses**	**X**	**X**	**X**
Software development planning	5.1.1; 5.1.2, 5.1.3; 5.1.6; 5.1.7; 5.1.8; 5.1.9	X	X	X
	5.1.5; 5.1.10; 5.1.11; 5.1.12		X	X
	5.1.4			X
Software requirements analysis	5.2.1; 5.2.2; 5.2.4; 5.2.5; 5.2.6	X	X	X
	5.2.3		X	X
Software architectural design	5.3.1; 5.3.2; 5.3.3; 5.3.4; 5.3.6		X	X
	5.3.5			X
Software detailed design	5.4.1		X	X
	5.4.2; 5.4.3; 5.4.4			X
Software unit implementation	5.5.1	X	X	X
	5.5.2; 5.5.3; 5.5.5		X	X
	5.5.4			X
Software integration and integration testing	All Requirements		X	X
Software system testing	All Requirements	X	X	X
Software release for utilization at a system level	5.8.1; 5.8.2; 5.8.4; 5.8.7; 5.8.8	X	X	X
	5.8.3; 5.8.5; 5.8.6		X	X
Software maintenance process	All Requirements	X	X	X
Risk Control measures	All Requirements		X	X
verification of risk control measures	All Requirements		X	X
Software risk management process	7.4.1	X	X	X
	7.4.2; 7.4.3		X	X
Software configuration management process	All Requirements	X	X	X
Software problem resolution process	All Requirements	X	X	X

3.2.1 Software development process

Software development planning:

The aim of this part is to reduce risk of failure during the development phase. In order to achieve this, a documented plan for software development is created. This plan also shows the communication and documentation processes for development. A development plan must be created and this is necessary for every medical device class. The plan describes processes to be used in the development, deliverables – documentation, results, tasks, traceability, configuration and problem solving.

Software requirements analysis:

Developer will bring out the software functional and capability requirements including inputs and outputs. Defines the interactions between systems and verifies software requirements.

Software architectural design (Only for Class B and C):

The software architectural design activity requires the manufacturer to define the major structural components of the software, their externally visible properties, and the relationships between them [21].

Software detailed design (Only for Class B and C):

In this step the architecture is defined into smaller units. If all the units are defined then a detailed design must be created, this design will specify how these units will interact with each other.

Software unit implementation:

This part is bringing detailed design into life, by writing code. Coding standards and methods should be used to aid in unit implementation. Application Security Verification Standard (ASVS) offers guidelines how to write code and test in in the appropriate manner.

Software integration and integration testing (Only for Class B and C):

Software integration testing focuses on the transfer of data and control across a software module's internal interfaces and external interfaces such as those associated with medical device hardware, operating systems, and third-party software applications and libraries. This activity requires the manufacturer to plan and execute integration of software units into ever larger aggregated software items, ultimately verifying that the resulting integrated system behaves as intended [21].

Software system testing:

The developer shall establish and perform a set of tests. Testing at the system level requires the manufacturer to verify the software's functionality by verifying that the requirements for the software have been successfully implemented. Testing will show if the software produces the expected outputs or not.

Software release for utilization at a system level:

The developer shall ensure that all software verification activities have been completed and the results have been evaluated before the software is released.

Software maintenance process:

Once the software is finalized software maintenance plan must be established. This plan must include how to conduct problem and modification analysis and set guidelines on how implement modifications.

Software risk management process

This includes:

- Analysis of software contributing to hazardous situations.
- Risk control measures.
- Verification of risk control measures.
- Risk management of software changes.

Software configuration management system

This includes:

- Configuration identification.
- Change control.
- Configuration status accounting.

Software problem resolution process

This includes:

- Prepare problem reports.
- Investigate the problem.
- Advise internal parties.
- Use change control process.
- Maintain records.
- Analyse problems for trends.
- Verify software problem resolution.
- Test documentation contents.

4 CALCULATIONS

Calculations are divided into two parts. First part will evaluate the potential financial effect of the change in the patient management flow. The second part will focus on usability testing calculations based on user testing results.

4.1 Arena Simulation

Software Arena Simulation was used in order to evaluate the difference between two different workflows. For simulation purposes a patient flow process was constructed, consisting of specific stages that every patient could move through.

4.1.1 Current patient flow

First simulation looked at the current patient management flow. The model consisted of 5 stages:

- Nurse visit
- Physician visit 1
- Physician visit 2
- Lab tests
- Home

All the patients entered the model in „nurse visit“ stage and the final stage for all patients was „home“. Depending on the situation, patient could move through all stages, or could drop off from any specific stage. This model is shown in Appendix 1.

4.1.2 Patient flow with HealthCode AI platform

The second simulation looked at the patient management flow when adding HealthCode AI platform into the process flow. The second simulation model consisted of 6 stages:

- Nurse visit
- E-consultation
- Physician visit 1
- Physician visit 2
- Lab tests
- Home

The main changes that HealthCode AI added was the possibility for E-consultation. All the patients entered from „nurse visit“ stage and the final stage for all patients was „home“. Depending on the situation, patient could move through all stages, or could drop off from any specific stage. This model is shown in Appendix 2.

4.1.3 Transition probabilities between stages and resource use

The transition probabilities between stages were collected as expert opinions from four Estonian physicians. The approximate values (drop off rate in each stage) were gathered from research papers and from four Estonian physicians who were interviewed about specific tasks. These values are show in Appendix 1 and 2.

4.1.4 Cohorts

Both these simulations had the same set of input data:

- 100 patients
- 1 physician
- 1 nurse

4.1.5 Drop-off points

When constructing the model, it was important to show at which point do patients get sent home, or in other words at which point the patient exits the simulation. These drop off points show how many patients will have one visit and how many will have two visits with the physician. The main difference between two models is that in model B there is a possibility for the nurse to do an E-consultation together with the physician, which allows to send additional patients home or directly to further analyses (lab tests).

4.1.6 Model output

The simulation evaluated how many physician and nurse visits will occur per 100 patients. From that data it is possible to evaluate the cost-efficiency for central payers. If the results of these models provide enough proof that solution provided by HealthCode AI is cost-saving then it is possible to evaluate which parts of the software must be developed first and which of those give the most effect.

4.1.7 Model A

Model A was constructed based the experts feedback and information found in research papers [22, 23] and Ministry of Social Affair's Healthcare system review [24]. Process flow of the model is shown in Appendix 3.

Model A describes the current patient management flow. In this model the patient can exit the flow in 6 events. First decision making has a 75% possibility that further information is required and the patient is asked to come in for a visit with physician or nurse. 40% of patients have visits with nurse and 60% of patients have visits with physician. Nurse sends about 10% of patients home after the appointment and 30% to have further analyses. Other patients are referred for a physician visit. Patients that are sent to analyses by nurse have 70% chance that after the results they do not need any more visits. 10% of patients are sent home after first visit with Physician. 70% are sent for further analyses and 70% from analyse results don't require any more visits.

The time it takes for consultation and for each physician visit is as follows: 5 minutes, 15-19 minutes and 7-15 minutes. The main difference in time between first and second visit is the information that physician has in hand before the patient steps into the physician office. This model was simulated 50 times to see how many users left each drop off point.

Model A results

The simulation (Figure 4.1) showed that on average 29.9 patients had visit with nurse. From those visits 17.7 patients were sent to 1st physician appointment and 2.6 where sent directly to home. In total there were 62,12 1st physician appointments. 24,9 patients had both 1st and 2nd visit and 2,9 patients had both nurse visit and 2nd visit with physician. The total mean value for nurse and physician visits was 119.96 for 100 patients.

Count	Average	Half Width	Minimum Average	Maximum Average
Call patient in from call	74.3600	1.39	65.0000	86.0000
From call to nurse appointment	29.9800	1.27	19.0000	41.0000
From call to physician I appointment	44.3800	1.20	35.0000	56.0000
From I phy app to analyses	43.7200	1.23	36.0000	52.0000
From phy I analyses to phy II app	12.7600	0.87	6.0000	20.0000
From Phy I app to II app	12.1400	1.02	4.0000	19.0000
Nurse analyses send to II physician appointment	2.9600	0.47	1.0000	8.0000
Nurse Analyses stay at home	6.6200	0.74	2.0000	12.0000
Nurse send to analyses	9.5800	0.83	3.0000	15.0000
Nurse sends to I physician appointment	17.7400	1.11	9.0000	26.0000
Physician analyses send home	30.9600	1.19	22.0000	37.0000
Physician appointment II send home	27.8600	1.29	20.0000	37.0000
Physician I send home	6.2600	0.75	1.0000	13.0000
Send Home Call	25.6400	1.39	14.0000	35.0000
Send home Nurse	2.6600	0.46	0.00	7.0000

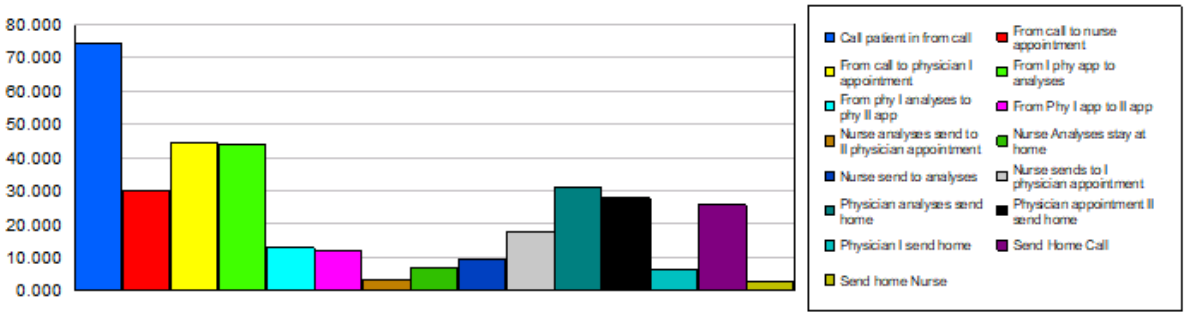


Figure 4.1 Model A simulation results for patient flow drop off points

4.1.8 Model B

Model B shows the workflow change when using ML to pre-question patients and provide nurses and physicians with decision making point. Research paper by Holman et al [22] proposed that gathering information from patient before they enter the clinic and preparing agenda could reduce the physician's and patients stress and also reduce medical errors. Pre-visit planning could help to reduce the instances of physicians having incomplete information on each patient and for the need for physicians to start looking for additional information during the visit. Medical experts say that giving patient information before hand to the physician can make the visit 60% more efficient for regular patients and 40% more efficient for patients with chronical disease. Experts proposed that with information beforehand unnecessary visits can be avoided and face to face visits can be faster as they have information which they can use to improve communication with patients. Model B patient flow is shown in Appendix 4.

Initial decision making is done after the nurse reviews the information provided by the patient. 30% of patients are instructed to stay home, 35% are called in for an appointment and in 35% of cases the nurse will make an E-Consultation with physician to decide on the next steps. 40% of patients that are called in have visit with nurse and 60% with physician. The next steps for both of these flows will remain the same as in model A. This is based on the opinion of physicians that were interviewed. E-consultation has 50% of chance that the patient will be invited in for an appointment or is sent to further analyses. What is important in this model is that patient who are sent directly to additional testing will only need to do 1 visit with physician. This model was simulated 50 times to see how many users left at each drop off point.

Model B results

This simulation (Figure 4.2) showed that mean value of 1st visits was 48,3 and 2nd visits was 26,3. Nurse had 16.7 visits. Total mean value of 1st visit + 2nd visit + nurse visit was 91.3. This means that for 100 patients there were less visits than there were patients. This is due to e-consultation option which led to patients having only one physician visit. Model A had an average of 119.96 visit total which is approximately 27 more visits per 100 patients.

Count	Average	Half Width	Minimum Average	Maximum Average
From E Consultation to analyses	15.6000	2.57	11.0000	22.0000
From E conusltation to I physician appointment	18.6000	2.55	15.0000	23.0000
From I physician appointment to II physician appointment	9.0000	2.52	6.0000	17.0000
From Leia call Patient in for appointment	36.8000	4.33	25.0000	44.0000
From Leia to Home	29.0000	4.07	21.0000	39.0000
From Physician analyses send to II appointment	16.2000	3.18	10.0000	24.0000
I physician appointment sent to analyses	34.7000	3.95	22.0000	42.0000
Number of E consultations	34.2000	3.07	27.0000	40.0000
Number of I physician appointments from Leia	20.1000	3.67	11.0000	28.0000
Number of nurse appointments from Leia	16.7000	2.84	10.0000	22.0000
Nurse analyses send to home	4.0000	1.17	2.0000	7.0000
Nurse send from analyses to II appointment	1.1000	0.71	0.00	3.0000
Nurse send to I physician appointment	9.6000	2.14	5.0000	15.0000
Physician analyses send home	34.1000	3.76	26.0000	43.0000
Physician appointment II send home	26.3000	3.63	19.0000	36.0000
Physician I send home	4.6000	2.06	0.00	9.0000
Send home Nurse	2.0000	1.12	0.00	5.0000

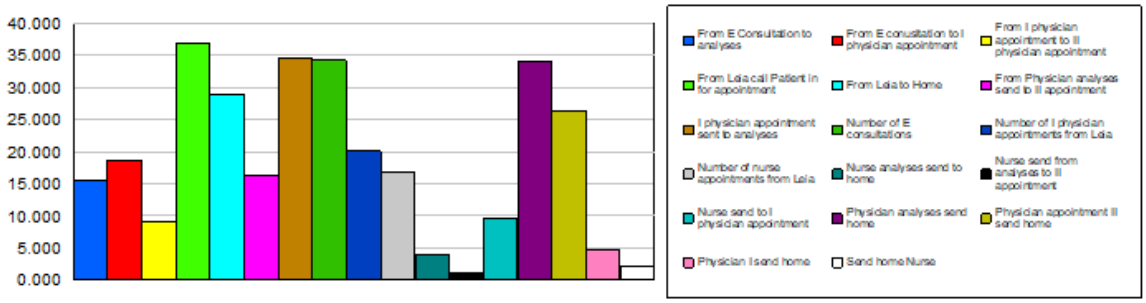


Figure 4.2 Model B Patient simulation results for patient flow drop off points

4.2 Economic values per visit

This paragraph evaluates the economic benefits of modifying the current patient flow for Estonian central payer. Estonian Health Insurance Fund's data from 2018 financial report was used as basis for calculations [25].

Table 6. Estonian Health Insurance Fund's data

Primary Care Physician	**2018**
Primary Care Physician financing (thousands €)	1 271 55
Total amount of visits	6 955 358

Dividing primary care physician financing with total amount of visits we get an estimation on how much does one physician visit cost.

$$Cost\ per\ one\ visit = \frac{Primary\ care\ physician\ financing}{Total\ amount\ of\ visits} = \frac{127155000\ €}{6955358\ Visits} = 18.28\ €\ per\ visit \quad (4.1)$$

Taking into consideration the results from models A and B it is possible to calculate the cost difference per 100 patients.

$$Model\ A\ cost\ per\ 100\ patients = number\ of\ visits * cost\ per\ one\ visit$$
$$= 120 * 18.28€ = 2193.6\ € \quad (4.2)$$

$$Model\ B\ cost\ per\ 100\ patients = number\ of\ visits * cost\ per\ one\ visit$$
$$= 91.3 * 18.28€ = 1668.9\ € \quad (4.3)$$

Cost difference will show the amount spent for every 100 patients. This amount can be scaled up and evaluated for the case of 1 million patients.

$$Cost\ Difference = Model\ A\ cost\ per\ 100\ patients - Model\ B\ cost\ per\ 100\ patients = 2193.6€ - 1668.9€ = 524.7€$$

(4.4)

The possible decrease of visits was evaluated while taking into account the current ratio of patients who had a visit and the total amount of visits.

$$visits\ per\ patient = \frac{Total\ amount\ of\ visits}{Number\ of\ patients\ who\ had\ a\ visit} = \frac{6955358}{1031449} = 6.74$$

(4.5)

Let's calculate the new ratio of visits per patient taking into account the change of visits per 100 patients. First of all, we must find the difference between 120 and 91,3 which is 76.08%. To find the new ratio we must see how much is 76.08% from the visits per patient 6.74. This new ratio value is 5.12.

Now it is possible to estimate the new number of visits with the ratio of 5.12.

$$New\ Total\ amount\ of\ visits = Number\ of\ patients\ who\ had\ a\ visit * new\ ratio = 1031449 * 5.12 = 5281018$$

(4.6)

Calculating the cost difference between total amount of visits ratios:

$$Cost\ difference\ between\ ratios = Primary\ Care\ Physician\ financing - (New\ total\ amount\ of\ visits\ * new\ ratio) = 127\ 155\ 000 - (5281018 * 18.28) = 30\ 617\ 990€$$

(4.7)

These preliminary calculations estimate that when bringing down the physician visit ratio from 6.74 to 5.12, the central payer would save approximately 30.6 million € per year.

One of the key elements in model B is that the patient fills in the health questionnaire using the software before contacting the physician centre. The second part of calculations follows up on the software usability. In the next chapter, different tasks of the software will be evaluated to gather information about which tasks are easy to understand for the user and which ones are not. Model B showed significant effectiveness, but this can only be achieved if the software has great usability.

4.3 Usability calculations

This part evaluates how well users were able to complete tasks on test versions of the software. These calculations give valuable feedback on how well the user interface was designed and which parts need more attention, additional development or redesign.

The data for the calculation was collected from 8 users. Each user was given the same information about the software; while they were testing the software there was no communication between tester and observer. The process was recorded by desktop recorder application and the results were analysed using Microsoft Excel.

The ISO 9241-11 standard defines usability as "the extent to which a product can be used by specified users to achieve specified goals with **effectiveness**, **efficiency** and **satisfaction** in a specified context of use [26]".

The ISO/IEC 9126-4 Metrics recommends that usability metrics should include:
Effectiveness: The accuracy and completeness with which users achieve specified goals.
Efficiency: The resources expended in relation to the accuracy and completeness with which users achieve goals.
Satisfaction: The comfort and acceptability of use
Users were testing the "Leia" software and they had nine tasks to complete. Observer looked at the time it took to complete each task and if the task was successfully completed or not. The tasks were as follows:
Task 1 – Enter symptom "fever"
Task 2 – Describe if your symptoms are getting worse or not
Task 3 – Choose between events
Task 4 – Entering symptoms
Task 5 – Accepting symptom
Task 6 – Describe symptom duration

Task 7 – Describe symptom details

Task 8 – Adding additional symptoms or not

Task 9 – Finalizing

Table 7. User testing results.

Task	User 1 / seconds	Y / N	User 2 / seconds	Y / N	User 3 / seconds	Y / N	User 4 / seconds	Y / N	User 5 / seconds	Y / N	User 6 / seconds	Y / N	User 7 / seconds	Y / N	User 8 / seconds	Y / N
1	31	Y	55	N	29	Y	49	Y	81	Y	120	Y	23	Y	36	Y
2	2	Y	5	Y	18	Y	6	Y	3	Y	38	Y	9	Y	10	Y
3	11	Y	7	Y	14	Y	10	Y	8	Y	63	Y	37	Y	10	Y
4	18	Y	24	Y	20	Y	22	Y	24	Y	94	Y	33	Y	17	Y
5	2	Y	-	-	-	-	-	-	-	-	-	-	2	N	11	N
6	14	Y	26	Y	40	Y	24	Y	20	Y	97	Y	30	Y	24	Y
7	57	Y	61	Y	97	Y	14	Y	11	Y	60	Y	12	N	45	Y
8	14	Y	8	Y	20	Y	13	Y	7	Y	120	Y	14	Y	11	Y
9	4	Y	3	Y	5	Y	5	Y	3	Y	30	Y	4	Y	15	Y

The ISO/IEC 9126-4 approach to Usability Metrics:

$$Effectiveness = \frac{Number\ of\ tasks\ completed\ successfully}{Total\ number\ of\ tasks\ undertaken} * 100\% \quad (4.8)$$

Effectiveness results for each task using the Effectiveness formula:

Table 8. Effectiveness results per task

Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9
88%	100%	100%	100%	33%	100%	88%	100%	100%

$$Timed\ Based\ Efficiency = \frac{\sum_{j=1}^{R} \sum_{i=1}^{N} \frac{n_{ij}}{t_{ij}}}{NR}$$

(4.9)

N = The total number of tasks (goals)

R = The number of users

n_{ij} = The result of task i by user j; if the user successfully completes the task, then N_{ij} = 1, if not, then N_{ij} = 0

t_{ij} = The time spent by user j to complete task i. If the task is not successfully completed, then time is measured till the moment the user quits the task

Table 9. Time based efficiency results using formula 4.9

Time based efficiency	Task1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9
Goals per second	0,022386	0,186623	0,084137	0,041433	0,166667	0,038983	0,030684	0,07961	0,208333
Goals per minute	1,30	11,20	5,04	2,48	10,00	2,33	1,84	4,77	12,50
Average time on task / Second	53	11,375	20	31,6	5	34,375	44,6	25,87	8,62
Time a task should take / Second	30	10	20	15	3	10	60	10	5

4.4 Conclusion of usability calculations

Before starting usability testing HealthCode AI set its own targets for completion of tasks and how much time each task should take. Task completion rate must be 100%; time per task is shown in Table 8. Usability test showed that some users were not able to complete all tasks. Problematic tasks were Task 1, 5 and 7. To understand why these tasks were not completed fully, interview was conducted with each user to better understand what was the problem. For example in Task 5, the

main problem was that the button „accept“ was not noticeable enough so users skipped it. This information led to the change of UI design and this is the reason why some users don’t have value in Task 5 row (Table 7), because these users were already using the updated version of the software. The effect can be seen in Task 4 as it is 100%. Time set per each task was only met for Task 3. This suggested that each and every task must be analysed more in order to understand what are the design faults and how they could be improved. For this reason, another UI version was created (design changes were made relating to each task). The effect of these changes must be evaluated by doing user testing once again.

5 SUMMARY

In the first part of the thesis an overview for medical device and software as a medical device was given. An overview of the healthcare industry's current situation suggested that there is a need for big data analytic tools. For example, in the U.S. it is predicted that healthcare data will reach yottabyte (1024 zettabytes) and this amount of data can only be analysed by using big data analytic tools.

The second chapter gives an overview of HealthCode AI OÜ smart device development. The device on hand is explained in this chapter with also the explanation why it categorizes under SaMD. In this chapter HealthCode AI MD development process steps were created, taking into account the requirements set by the MDR. Necessary MDR requirements for software development were described and HealthCode AI software development method was created. This method combines Agile development principles and Quality Management System requirements. The goal of this chapter was to apply MDR requirements to smart device development and show how this was accomplished from project management point of view. Example cases were given to show how customer requirements were transformed into functions and then prioritised for development.

The third chapter focused on guidelines for developing MD Class IIa Software. Process flows were created for CE marking process, Quality Management System and Software development life cycle. These flows are shown in Appendix 5, 6 and 7. These processes flows were generated by researching the requirements set out in MDR and in ISO 13485 and IEC 62304 standard.

In the fourth chapter two patient flow models were simulated by using Arena Simulation. The first model describes the current patient management flow and the second one describes HealthCode AI proposed patient management flow. Both simulations were done with same input data and the result was that HealthCode AI proposed model has 27 less visits per 100 patients, which means less burden for the primary care setting. This result was used to calculate cost-savings for Estonian central payer Estonian Health Insurance Fund (Eesti Haigekassa). The cost saving was 30.6 million euros per year. This proves that solution offered by HealthCode AI has statistical benefits in terms of reducing resource use burden in the primary care setting and increasing cost-efficiency for the central payer.

The next part of the calculations focused on usability calculations. Usability testing was done with 8 users who were all given the same tasks. The results showed that all users were able to complete 6 out of 9 tasks; Task 5 had the lowest 33% completion rate. Time based efficiency was calculated which showed that users were completing tasks but taking too much time. This data gives insight for project manager to notice which Tasks are problematic and need change in either design or in functionality.

Additional research need

In current modelling the change in resource use of physician/nurse time was not calculated, nor the change in consultation structure (increase in face to face time with patient). In future, a Markov model could be used to calculate change in these resources. This model would also allow to calculate drop off point results and compare them to results from this thesis. In case long-term data from HealthCode AI system use will be available, health economic calculation about additional health impact (decrease in specific health events; prevention efficacy, treatment efficacy) could be calculated.

KOKKUVÕTE

Lõputöö Esimene osa annab ülevaate meditsiiniseadmest ja tarkvarast kui meditsiiniseadmest. Hetke olukord tervishoius näitab vajadust tarkvara järele, mis on võimeline analüüsima suuremahulisi andmekogumeid. Näiteks Ameerika Ühendriikides eeldatakse, et tervishoiu andmemaht jõuab varsti *yottabyte*'ni ning sellist andmemahu suurust on võimalik ainult analüüsida kasutades suurandmete analüüsitarkvara.

Teine osa tööst annab ülevaate firma HealthCode AI OÜ nutika meditsiiniseadme arendamisest. Antud seadmest antakse ülevaade ja kirjeldatakse, miks see kategoriseerub tarkvara kui meditsiiniseadme alla. Selles peatükis koostatakse ka HealthCode AI meditsiiniseadme arendamise protsessid, võttes arvesse meditsiiniseadme regulatsioone. Välja on toodud vajalikud nõuded tarkvara arendamiseks meditsiiniseadme regulatsioonidest ning luuakse HealthCode AI tarkvara arendamise meetod. See meetod kombineerib Agiilse arenduse printsiibid ja kvaliteedi juhtimissüsteemi nõued. Selle peatüki eesmärk on juurutada meditsiiniseadme regulatsioonide nõudeid nutika seadme arendamisesse ning näidata, kuidas seda viidi ellu projektijuhtimise perspektiivist. Tuuakse välja näidis, kuidas kliendi nõuded tõlgendati funktsioonideks ja määrati tarkvaraarenduse prioriteedid.

Kolmas peatükk keskendub Klass IIa tarkvaralise meditsiiniseadme arendamise juhiste koostamisele. Loodud on protsessivood CE märgise saamiseks, kvaliteedijuhtimissüsteemi juurutamiseks ning tarkvara elu-tsükli loomiseks. Need vood on toodud välja Lisades 5, 6 ja 7. Antud vood põhinevad meditsiiniseadme regulatsioonidele ning ISO 13485 ja IEC 62304 standardites esitatud nõuetele.

Neljandas peatükis simuleeritakse kahte patsiendi manageerimise voogu kasutades programmi Arena Simulation. Esimene mudel kirjeldab hetke seisu patsientide manageerimisel ning teine mudel kirjeldab HealthCode AI poolt pakutavat patsiendi manageerimise voogu. Mõlemad simulatsioonid viidi läbi samade sisendväärtustega ja tulemuseks oli, et HealthCode AI poolt pakutavas mudelis esines 100 patsiendi kohta 27 viisti vähem kui teises mudelis. Tulemus viitab sellele, et HealthCode mudeliga väheneb koormus esmatasandi tervishoiuressursidele. Simulatsioonidest saadud tulemuste põhjal arvutati välja kulutõhusus Eesti Haigekassale. Kokkuhoid kasutades HealthCode AI mudelit oli 30,6 millionit eurot aastas. Antud mudel tõestas,

et HealthCode AI poolt pakutav patsiendi manageerimise voog on statistiliselt kasumillik esmatasandi arstiabi koormuse vähendamiseks ning võib suurendab Eesti Haigekassa kulutõhusust.

Järgmine arvutuslik osa keskendus kasutusmugavus kalkulatsioonidele. Kasutusmugavuse testimine viidi läbi 8 kasutajaga, kellele kõigile anti samad ülesanded. Tulemused näitasid, et kõik kasutajad olid võimelised lõpetama 6 ülesannet 9-st; Ülesandel 5 oli kõige madalam õnnestumise protsent (33%). Arvutati ajakulu effektiivsus, mis näitas, et kasutajad said ülesannete täitmisega hakkama, kuid see võttis liiga kaua aega. Need tulemused annavad sisendid projektijuhtile, milliste ülesannetega on vaja ellu viia kas disaini või funktsionaalsuse muudatusi.

Täiendava uurimustöö võimalused

Käesoleva töö modelleerimises ei uuritud muutusi arsti- ja õevisiidi resurssikasutuses ega muutusi konsultatsiooni struktuuris (täiendav näost-näkku suhtluse aeg patsiendiga). Tulevastes uurimustes võiks kasutada Markovi mudelit, et arvutada välja nende resurssidega seotud muutused. Selline mudel saaks samuti arvutada välja patsiendi väljumiskohad mudelist ning võrrelda neid käesoleva lõputöö tulemustega. Kui HealthCode AI pikaajalised andmed muutuvad kättesaadavaks, on võimalik viia läbi terviseökonoomilised kalkulatsioonid täiendava tervisemõju kohta (nt langus teatud tervisesündmuste esinemissageduses; preventsioonitegeevuste tõhusus; ravi efektiivsus).

6 REFERENCES

1. Sterling, Leon S. The Art of Agent-Oriented Modeling. London : The MIT Press, 2009.

2. European comission https://ec.europa.eu/growth/sectors/medical-devices/regulatory-framework_en (accessed 15.03.2019)

3. MedCert homepage https://www.medcert.de/wp-content/uploads/files/MDR_new_conformity_assessment_procedures/MDR-Conformity-Assessment-Procedure-Class-IIa.pdf (accessed 15.03.2019)

4. Nishita Mehta. Concurrence of Big Data Analytics and Healthcare: A Systematic Review, 2018

5. W Raghupathi. Big data analytics in healthcare: promise and potential, 2014

6. Robert M. Kaplan, Dominick L. Frosch. Decision making in medicine and health care, 2004

7. Robert Pearl, M.D. 5 Things Preventing Technology Adoption In Health Care, 2014

8. ISO 62304:2006. https://www.iso.org/obp/ui/#iso:std:iec:62304:ed-1:v1:en (accessed 15.04.2019)

9. ISO 13485:2016. https://www.iso.org/obp/ui/#iso:std:iso:13485:ed-3:v1:en (accessed 15.04.2019)

10. S.Balaji. Wateerfall vs V-model vs Agile: a comparative study on SDLC, 2012

11. P Beynon-Davies, C Carne, H Mackay & D Tudhope. Rapid application development (RAD): an empirical review, 2017

12. Agile manifesto https://agilemanifesto.org/ (accessed 15.04.2019)

13. Helen Sharp, Tracy Hall. Agile Processes in Software engineering and Extreme programming, 17th international Conference. XP 2016, p 132

14.Certification Experts https://certification-experts.com/notified-body-accreditation-for-medical-device-regulation-eu-2017-745/ (accessed 16/04/2019)

15. Hugh Harvey. https://towardsdatascience.com/how-to-get-clinical-ai-tech-approved-by-regulators-fa16dfa1983b , 2017 (accessed 16/04/2019)

16. MedCert homepage https://www.med-cert.com/en_certification/en_your-checklist/en_technical-file/ (accessed 17/04/2019)

17. International medical device regulators forum http://www.imdrf.org/docs/imdrf/final/technical/imdrf-tech-170921-samd-n41-clinical-evaluation_1.pdf (accessed 17/04/2019)

18. Clinical trials arena homepage https://www.clinicaltrialsarena.com/comment/clinical-evaluation-for-software-as-a-medical-device/ (accessed 17/04/2019)

19. International medical device regulators forum http://www.imdrf.org/docs/imdrf/final/technical/imdrf-tech-131209-samd-key-definitions-140901.pdf (accessed 17/04/2019)

20. International medical device regulators forum http://www.imdrf.org/docs/ghtf/final/sg5/technical-docs/ghtf-sg5-n2r8-2007-clinical-evaluation-070501.pdf (accessed 17/04/2019)

21.Medical Design Brief https://www.medicaldesignbriefs.com/component/content/article/mdb/features/articles/27722?start=1 (accessed 23/04/2019)

22. G Talley Holman John W Beasley Ben-Tzion Karsh Jamie A Stone Paul D SmithTosha B Wetterneck. The myth of standardized workflow in primary care, 2015 / WWW: https://academic.oup.com/jamia/article/23/1/29/2380110

23. José Joaquín Mira. Patient report on information given, consultation time and safety in primary care, 2010

24.Sotsiaalministeerium. https://www.sm.ee/sites/default/files/content-editors/Tervishoid/hit_-_estonia_-_web_version_01.06.2018.pdf (accessed06.05.2019)

25. Haigekassa. https://www.haigekassa.ee/sites/default/files/uuringud_aruanded/Eesti%20Haigekassa_majandusaruanne_2018.pdf (accessed 06.05.2019)

26. Usability Geek, https://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/ (accessed 06.05.2019)
27. Synopsys homepage https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/ (accessed 06.05.2019)

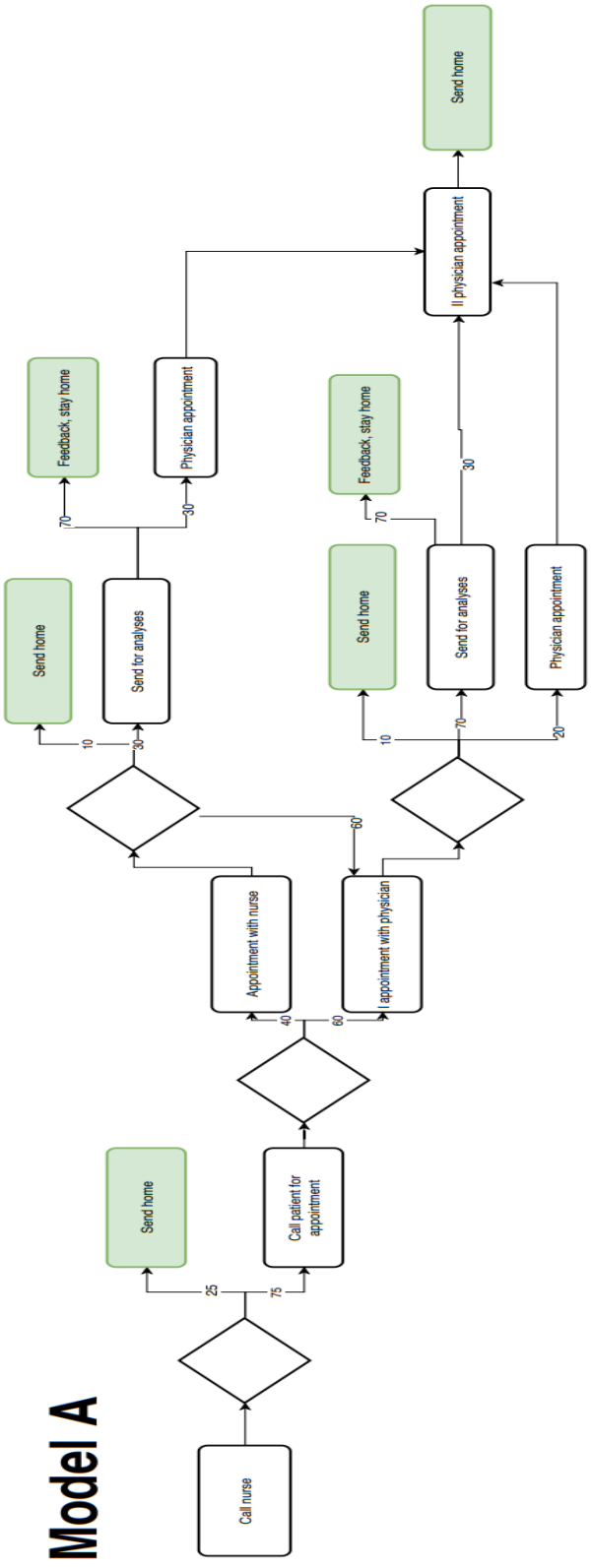
28. Justin Altschuler. Estimating a Reasonable Patient Panel Size for Primary Care Physicians With Team-Based Task Delegation, 2012

29. Char DS, Shah NH, Magnus D. Implementing machine learning in health care - addressing ethical challenges. N Engl J Med. 2018;378:981–983. doi: 10.1056/NEJMp1714229.

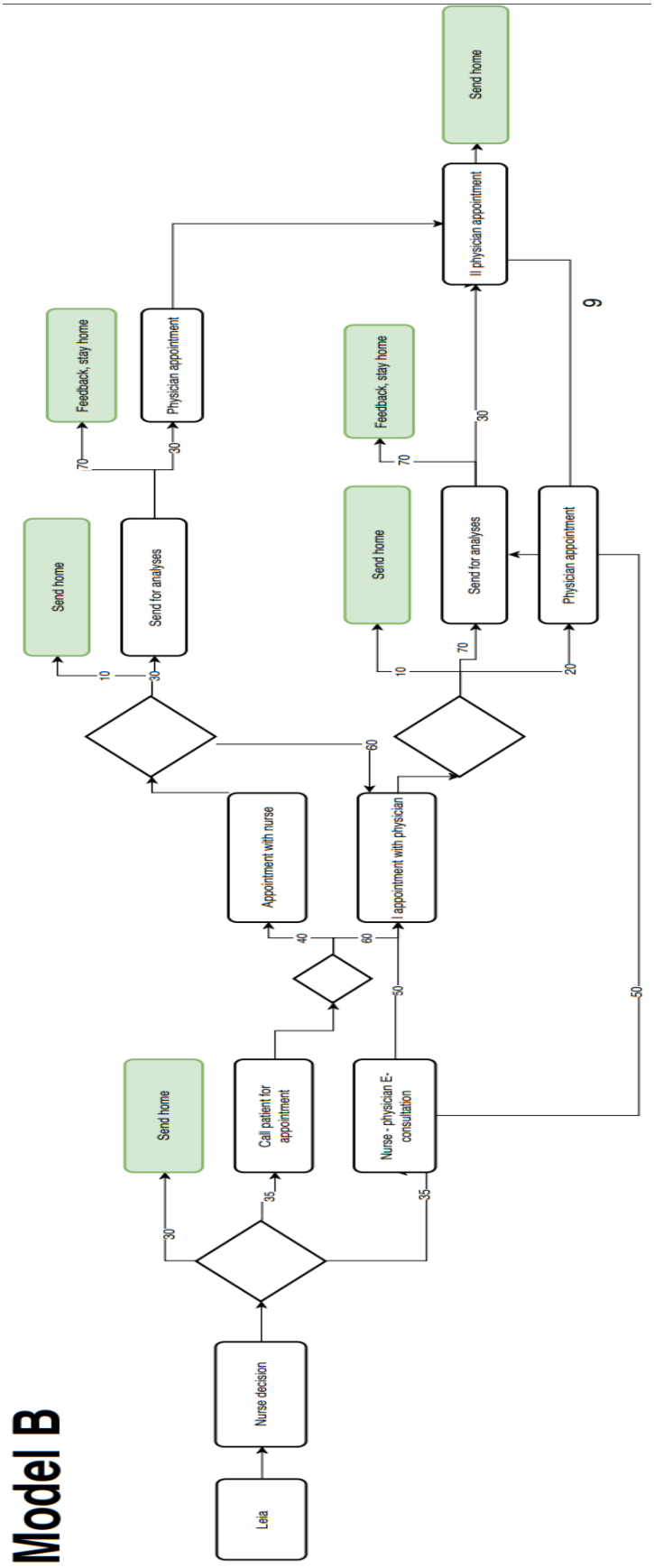
30. Angwin J, Larson J, Mattu S, Kirchner L (2016) Machine Bias. Available via http://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

APPENDICES

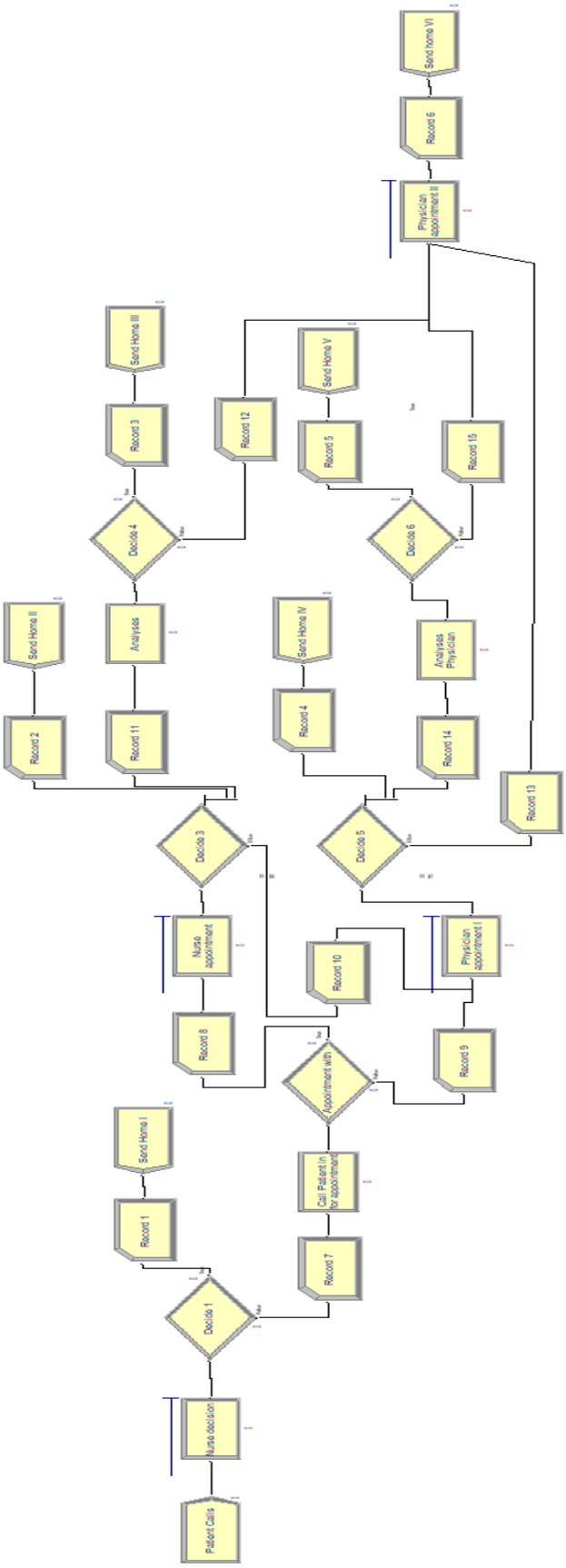
Appendix 1 Model A patient management flow



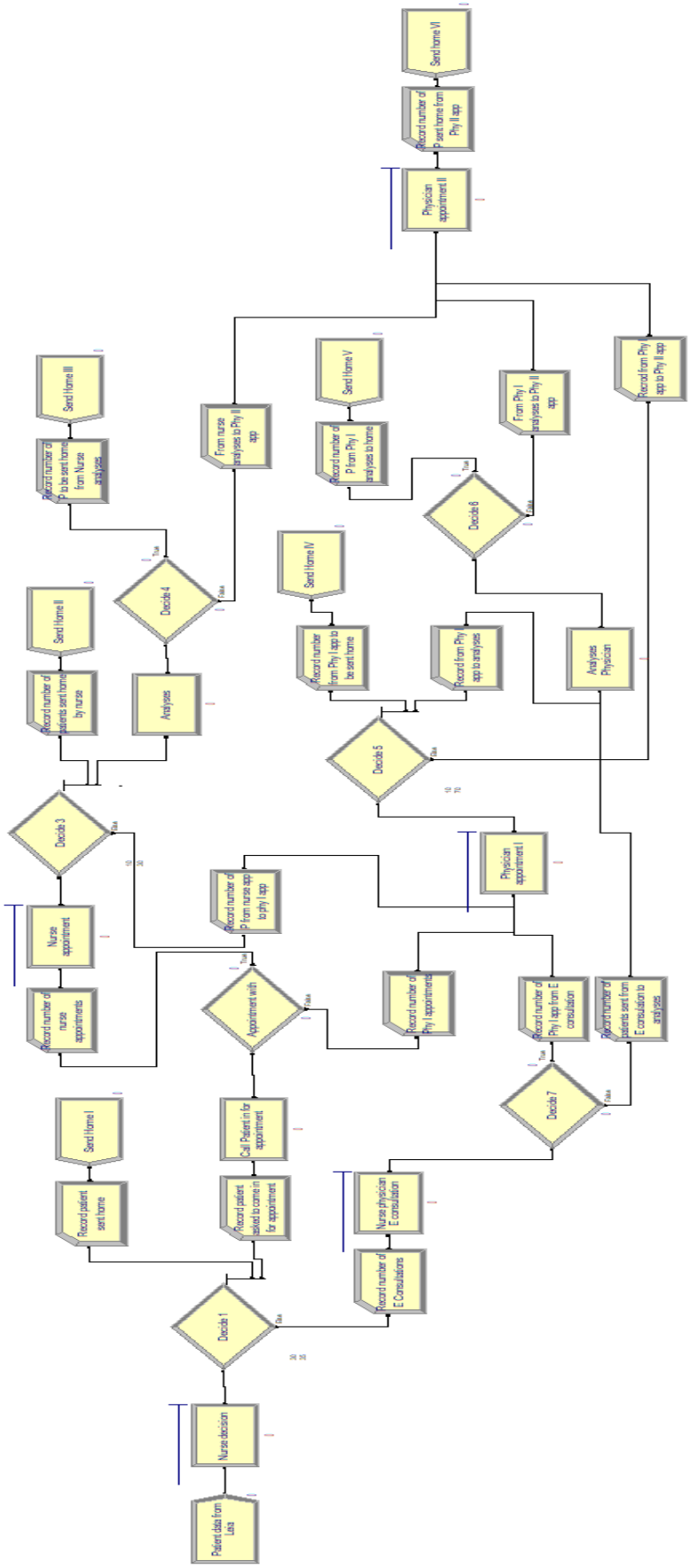
Appendix 2 Model B patient management flow



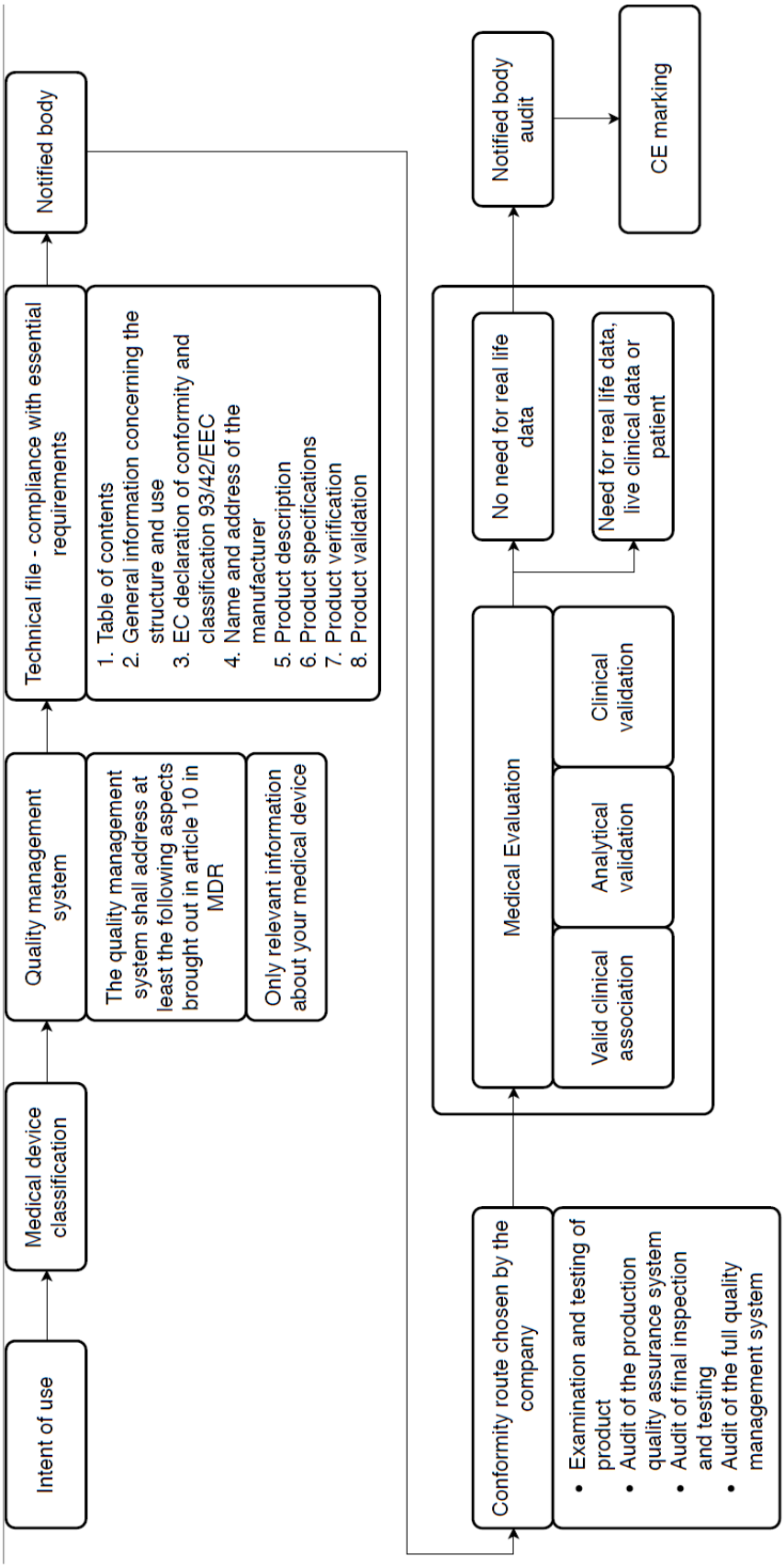
Appendix 3 Model A – Arena Simulation Model



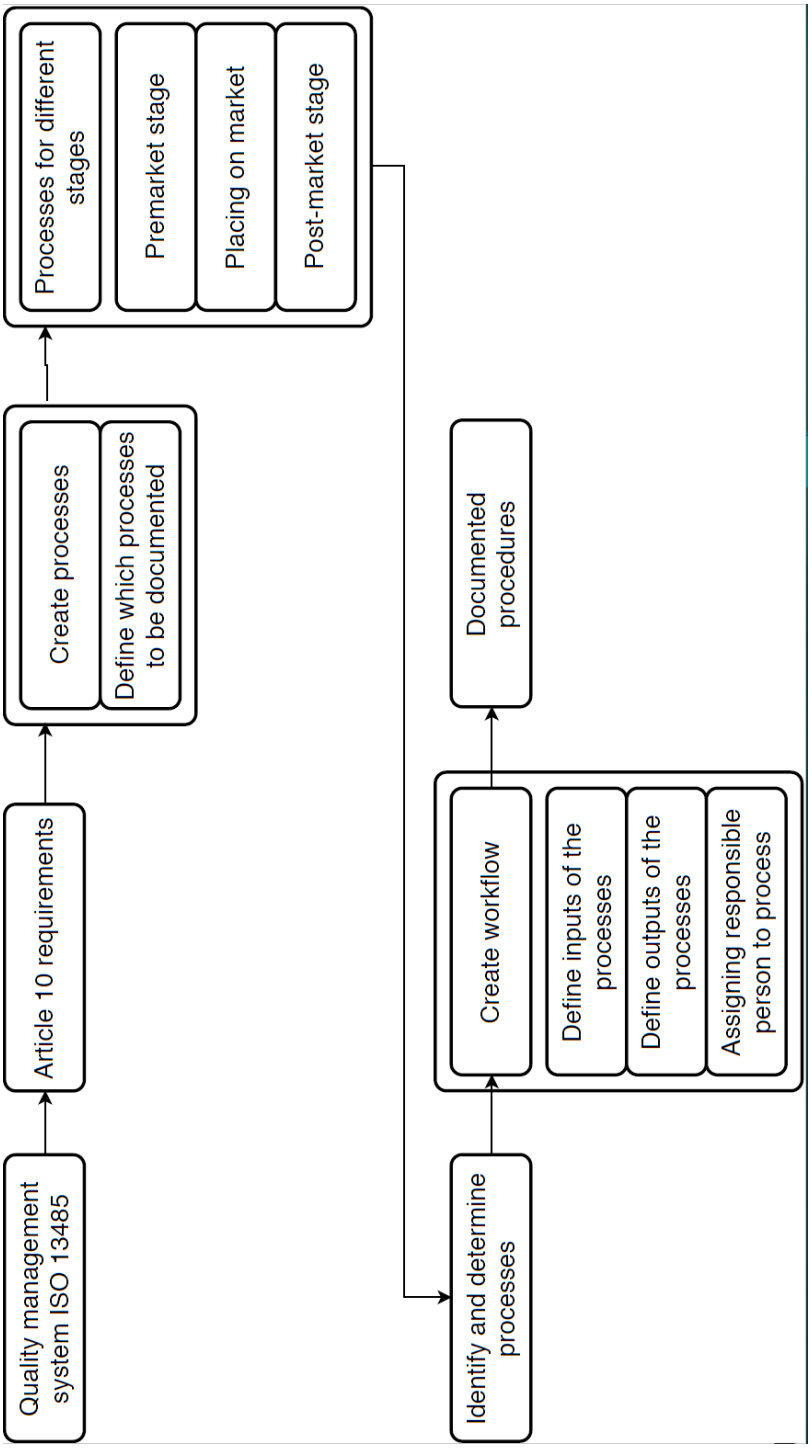
Appendix 4 Model B – Arena Simulation Model



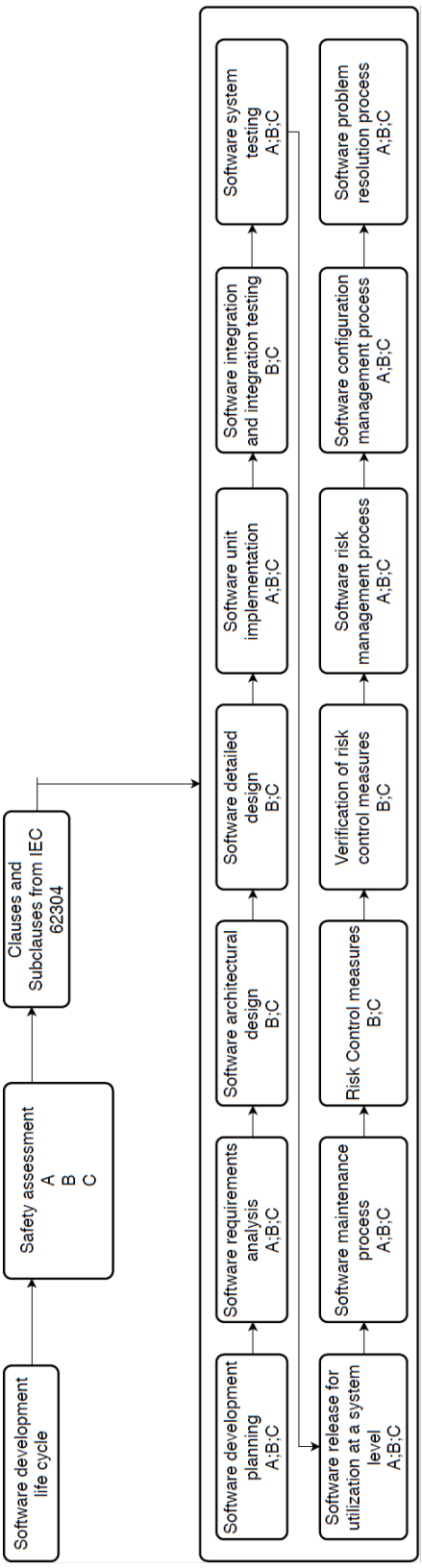
Appendix 5 CE Process flow



Appendix 6 Quality Management System flow



Appendix 7 Software development life cycle



Appendix 8 Change in patient management flow with HealthCode AI