The Gap between Visualization Research and Visualization Software (VisGap) (2020)
C. Gillmann, M. Krone, G. Reina, T. Wischgoll (Editors)

Towards Closing the Gap of Medical Visualization Research and Clinical Daily Routine

Robin G. C. Maack[1], Dorothee Saur[2], Hans Hagen[1], Gerik Scheuermann [3] and Christina Gillmann [3]

[1] University of Kaiserslautern [2] Leipzig University, Medical Centre [3] Leipzig University, Institute of Computer Science

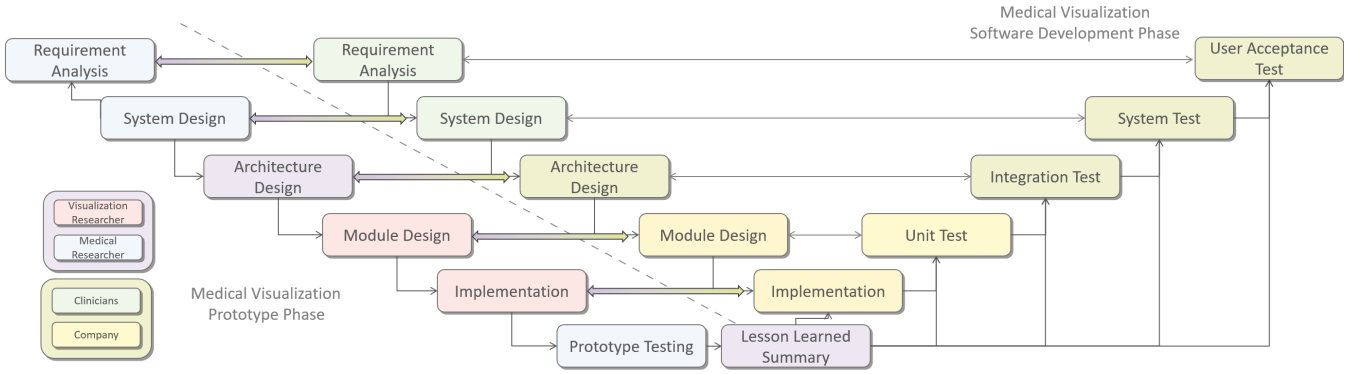

Figure 1: *Proposed workflow for a more efficient software development in medical visualization software involving all actors. The workflow is based on the V-Model which is the basis of medical software development. Phases are color coded according to the actor(s) that are involved.*

Abstract

Medical visualization papers are constantly published throughout the last years, but many never make their way into clinical daily routine. In this manuscript we aim to examine the gap between visualization research and clinical daily routine and suggest a mechanism that can lead towards closing this gap. We first identify the actors involved in developing new medical visualization approaches and their different views in this process. Then we develop a software development process unifying all actors and their needs. In addition, we collect further barriers in the medical software development process.

CCS Concepts

• ***Human-centered computing*** → *Visualization theory, concepts and paradigms;* • ***Software and its engineering*** → ***Software verification and validation;***

1. Introduction

Medicine as an application in visualization has a long tradition, as the nature of medical images imply the need for suitable visualization techniques to review them [Chi01]. Medical visualization techniques are constantly published [Cla], but still the slice-by-slice reviewing technique is one of the most used in clinical daily routine [BSV*18]. Here, medical doctors scroll through a stack of medical images, reviewing one image at a single time and creating a three-dimensional image in their mind.

Here, the question arises, why are there so many medical visualization papers published, but they barely make their way into clinical daily routine? The reasons for this are manifold. First, medical visualization counts as a medical device. Thus, it falls under the European medical device regulation [Cou17b] (MDR), that describes requirements for medical devices in detail and describes development steps as shown in Section 2. Second, bringing a medical visualization to market involves a variety of parties, including researchers, companies and medical doctors. All these groups have different goals when working with medical visualizations, but it is hard to find intersections in their interests. This problem is reviewed in Section 3. At last, contrary to other applications, there exists a large set of requirements that need to be fulfilled in order to create a successful medical visualization. These requirements orig-

DOI: 10.2312/visgap.20201107

inate from the different actors involved in medical visualization development and their importance varies depending on which actor is asked, as Section 4 shows. To solve this problem, this paper examines the requirements for medical visualization and differentiates the importance of each actor in the medical visualization process to solve the requirement. Based on this, we aim to clarify the given software development process by MDR, while giving an overview of the current process in regard of software. We summarize the responsibilities of each actor, the dependencies when responsibilities change throughout the software development process and propose suitable intersections between the actor groups in Section 5, based on the V-Model software development strategy.

Based on the refined software development process, we aim to discuss open problems in Section 6. Section 7 will conclude this work.

Therefore, this paper contributes:

- A summary of the Status Quo in Medical Software Development
- A list of actors in medical software development
- Requirements for medical software development
- Refined software development process for medical software
- Open Problems in medical software development

2. Status Quo

In order to understand the gap between medical visualization research and clinical daily routine, we will review the currently used slice-by-slice reviewing technique for volumetric data sets acquired by MRI or CT, as well as the current state of the art in medical visualization research. In addition, we will summarize the MDR that forms the legal frame for developing medical visualization software.

2.1. Slice-by-slice vs. Medical Research

Slice-by-slice reviewing is a visualization often used in clinical daily routine [GH02]. Here, doctors scroll through a stack of slices in a scan of the patients' body. At one single time, they solely review one slice of the image. Medical doctors learned how to create a three-dimensional image in their mind. Although this method allows the pure and undistorted view on the original data, it is not able to allow any further analytic visualization of the dataset. This is, where medical visualization research comes into play.

Medical visualization research is a highly active field, growing so big, that subareas can be covered in surveys to obtain an overview in this area [Bis18, LSBP18, ROJP11]. Although researchers are aware of the presented gap [GAH*17, GS14], they often can't bring their software into production. In this context, Botha et al. [BPK*12] formulated open challenges when researching medical visualizations in 2012. The following table shows the complete list, along with the number of papers published in this area until the end of 2019.

The data was retrieved by the Web of Science [Cla], a web application that is able to filter research papers according to given key words. We do not consider this list as complete, as keywords may be phrased slightly different or synonyms exist. But the numbers in Table 2.1 give a lower bound for the research papers available in the respective topic and show, that all topics are actively researched.

Medical Visualization	2482
Heterogeneous Display and Computing Devices	39
Interactive Image Segmentation	218
Topological Methods	8
Integration of Simulation Models	136
Mappings and Reformations	101
Hyper-realism	5
Visual Analysis in Healthcare	424
Population Imaging	24

Table 1: *Number of available research approaches for the open challenges in medical visualization formulated in 2012 by Botha et al. [BPK*12].*

Still, these publications have not made their way into clinical daily routine. To prove this statement we examined the portfolio of the most important medical visualization companies on the market. We utilized Capterra [Cap], a website that lists companies in different applications. We selected companies, that offer medical visualizations, leading to a result of 18 companies in this area. We browsed their portfolio and identified that none of these companies provide suitable solutions for the formulated open challenges.

In addition to capterra, we searched for further companies offering visualization software in the medical domain. Here, a major observation can be made. Companies as Philipps [Kon04], TIPHCe [Agf20] and intrasense [Int18] offer products on data management systems. Recently, these data management systems are extended in order to inccorporate machine learning approaches [Ter20].

In addition to data management systems, visualization systems are offered by big medical software development companies as well. Phillips offers the IntelliSpace Portal 11 system [Kon20], a workspace for data analysis and visualization for a variety of medical applications. GE Health offers a visualization software for vascular, cardological and neurological visualization problems [GE 20]. These are some prominent examples of a group of companies selling visualization software for medical purposes. Although these companies offer a large variety in advanced visualization solution for the medical market, there are multiple factors, that can inhibit the use of such a software in clinical daily routine. First, clinicians require a training phase for new software, which is time that they usually lack. Second, the products come with a high cost, that clinics (especially smaller ones) can often not afford. Third, even if a clinic is willing to pay for a new software product it needs to be run on the already existing hardware in clinics. Here, clinics often lack a proper hardware setting.

This clearly indicates the gap between visualization research and clinical application of new medical visualization techniques.

The source of this gap is very manifold. Besides different actors involved in medical visualization software development and varying requirements, the definition of medical software itself contributes to the gap. One very dramatic reason for this gap is in fact

that medical visualizations count as medical devices and therefore fall under the medical device regulation.

2.2. Medical Device Regulation

In order to understand the legal restrictions that exist for medical software, this section summarizes the most crucial points of important legal texts. Please note, that this section gives an overview about the most critical points according to the view of a visualization researcher.

Medical Software in the EU is regulated by the Regulation (EU) 2017/745 MDR [Cou17b](Medical Device Regulation) and Regulation (EU) 2017/746 IVDR [Cou17c] (In Vitro Diagnostic Device Regulation). These instructions, defining what medical and similar devices are, were updated in 2017 to ensure higher safety standards. The core ideas include, that:

1. Clinical trials now have to take place in more than one EU member state while being evaluated only once in a coordinated manner.
2. The European database will save more information about medical devices, achieving a better acceptance by the public.
3. More information shall be granted to patients yielding intelligible processes.
4. The registration of devices can now be conducted for the whole EU instead of each member state on its own.
5. Compensations for patients have to be clear, in case a patient is harmed by a medical device.

2.2.1. Definition of Medical devices in terms of software

The MDR and IVDR clearly state a distinction between Medical Devices and In Vitro Diagnostic Devices, as each of them is regulated by a different document. The MDR governs how medical devices have to be registered in the EU. Article 2(1) of the MDR [Cou17b] states, that any software is a 'medical device' if 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.
- control or support of conception.

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.

Similar rules hold for 'In vitro diagnostic medical devices', where In vitro('in the glass') means, studies and procedures which are conducted outside the body in a controlled environment. Therefor a software, according to Article 2(2) [Cou17c], is such a device if used alone or in combination, intended by the manufacturer to be used in vitro for the examination of specimens, including blood and tissue donations, derived from the human body, solely or principally for the purpose of providing information on one or more of the following:

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

This implies, that all visualization software intended to be used in medicine is restricted by the given laws.

It is specified that Software as well as 'In vitro diagnostic medical devices' are also 'active medical devices', as they depend on a source of energy, other than that generated by a human itself, which also needs to be considered in the software development process.

2.2.2. Class of a medical device software

Medical devices and In vitro diagnostic medical devices are categorized into classes depending on intended purpose of the device as defined by Rule 11 in Annex VIII [Cou17b]. The classes correspond to the potential risk to a patient, in the context of the MDR, where class I can be non-invasive devices, which do not penetrate the human body, while class III devices might monitor or correct a defect of the heart. The classes IIa and IIb describe devices for therapeutic purposes and diagnosis, as long as decisions made with the help of those devices can not cause death or an irreversible deterioration of a person's state of health. However, if the decision may only cause a serious deterioration of a person's state of health or a surgical intervention, it can be classified IIb.

As the medical software might be used to control or guide hardware solutions, it can be assigned to a lower class as the hardware itself. Rule 3.3 in Annex VIII of the MDR therefor states that Software, which drives a device or influences the use of a device, shall fall within the same class as the device. If the software is independent of any other device, it shall be classified in its own right. Other rules of Annex VIII can also influence decisions on those devices, but rule 3.3 and 11 are the most important regarding medical device software.

The classification of the software has consequences when it comes to the registration of such a product, e.g. the period of document updates, since a class I medical device has less regulations than a class III product.

2.2.3. General Safety and Performance Requirements

For the successful registration of a medical software product, it is important to comply with the General Safety and Performance Requirements, as stated in the MDR Annex I. The product has to be completely evaluated it terms of each of the requirements. It should also comply with current state-of-the-art techniques, even if the term state-of-the-art was not directly defined by the MDR. Therefore harmonised standards published with the MDR shall be used to evaluate the usefulness of the medical software, if it describes a similar behaviour as the proposed medical device. These

harmonized standards are still being developed and will be added to the MDR as soon as they are ready.

Summarizing the most important rules of MDR Annex I we can formulate general statements about the design of medical software. First, the intended design of the software and the implementation should align, making the product save to use while minimizing risks. The minimization of risks should be planned in such a way that the usefulness does not negatively effect the benefit-risk ratio. Furthermore a risk management system has to be established and documented, which will be updated throughout the life cycle of the product, giving technical knowledge, education and training material to the users.

Measuring functions in software have to be designed in a way that they provide sufficient accuracy, precision, and stability for the use-case they are intended for. Limits of the measuring techniques have to be indicated and standardized Units have to be used, as defined by the Council Directive 80/181/EEC [Cou17a].

Software, according to the MDR, has to be designed in such a way that results can repeatedly and reliably be reproduced, while ensuring sufficient performance for the intended use case. Minimum requirements for the hardware have to be made by the manufacturer and security features, preventing loss and illegal access of data, have to be implemented. When monitoring a patient's health status additional alerts have to be embedded into the system, warning him about critical health situations. Additionally, on mobile platforms, apps have to be made with the available screen size, interaction styles and performance in mind.

3. Actors of medical Software Development

Developing medical visualization software requires different actors that need to work together. Actors in medical software development are:

- Researchers/ Research Institutions
- Companies
- Clinicians
- Doctors in medical research
- Governmental agencies

Figure 2 shows the four groups of actors that are involved in the medical visualization software development process. The strength of connection between these groups is indicated by the opacity of the connecting arrows. We determined the strengths of connections by questioning individuals from each actor group how close they are working together with the other groups. Although this is not based on a statistic significant number of statements, the actors ensured us that their opinion represents most of their coworkers opinions. The group of actors, their aims in medical visualization software development, workflows, challenges, and interfaces with the remaining groups are shown and will be explained below.

3.1. Visualization Researcher/ Visualization Research Institute

A researcher or research institute aims to make a scientific contribution in medical visualization. Researchers aim to identify an unsolved or insufficiently solved problem in medical visualization and find a proper solution. Here, they develop general concepts to solve a problem and utilize prototypes in order to show their applicability. Their challenges mostly arise in the prototype stage, where suitable solutions need to be found. After a successful prototype was created, usually the work of a visualization researcher ends. Visualization researchers are not paid to create and sell products and even if they do, it is lowly appriciated.

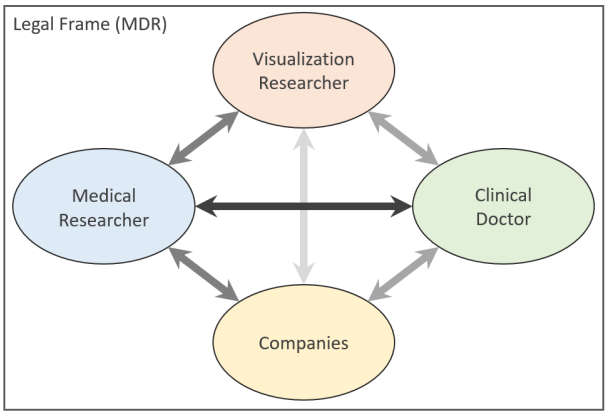

Figure 2: *Actors involved in the development of medical visualization software. They include visualization researcher, clinical doctors, medical researchers, and companies. The strength of their connection is indicated by the opacity of the connecting arrows.*

Researchers usually interact with medical researchers that provide them with exemplary datasets and research questions. They also collaborate with clinical doctors, but usually their first interaction with medical doctors is on the research site and their prototypes are not in a stage that can be utilized in clinical daily routine. Visualization researchers barely interact with companies. This is due to the problem, that visualization researchers need to publish their work in order to obtain scientific reputation. Contrary to this, a company tries to keep developments as secrets in order to maximize profits.

3.2. Companies

Companies in the area of medical visualization aim to offer ready to use products that can be sold to clinics. Their goal is to develop a unique software solution that can be sold to a variety of clients assisting during their daily tasks. Resulting from that, making scientific contributions is a minor goal to them. Companies are more concerned about the legal frame they need to hold, while still obtaining enough profit when selling products.

Companies strongly interact with medical researchers. In many cases they hire them to let them proceed the clinical trial stage or test software that is developed in the company. Companies also interact with medical doctors working in clinical daily routine. They advertise their software and aim to interview unsolved problems that they can implement. Companies are also open to work with visualization experts, but they wish to not directly publish

novel developments as their existence is depending on earning money. In cases a collaboration between companies and visualization researchers are build, contracts are made that keep researchers from publishing results for a specific time frame. This is probably the most important reason why companies and visualization researchers lack in collaborations.

3.3. Doctors in clinical daily routine

Doctors in clinical daily routine are not directly involved in the software development process of medical visualization software. Their important task is to define the requirements that software products need to fulfill. If a visualization product does not fulfill their needs they will not use it, which makes them the driving force in medical visualization development. Unfortunately, medicine is a very traditional field where data interpretation is taught in a very specific way usually related to the slice-by-slice reviewing technique. Medical doctors in clinical daily routine need to make decisions really quick, usually in a frame of minutes and therefore require a reliable review mechanism for their data. Although they might be open to learn novel review techniques, they simply lack the time to use algorithms that they do not fully understand or which require a lot of time to learn.

Clinicians highly interact with medical researchers and are in fact sometimes both. Depending on the role they have at a specific time, their interest vary. Their interaction with visualization researchers is rare, simply due to time restrictions and the lack of applicability of prototypes. The interaction with companies is given, but given by the software product that was chosen by the institution they are working for.

3.4. Medical Researcher

Medical researchers aim to improve medical knowledge and enhance routines that are used for diagnosis or treatment. The use of improved medical visualization is a major goal in their research [AA19]. Their role in the development of medical visualization software is to formulate requirements and make first tests of applicability for medical visualizations. In addition, they use advanced visualization techniques to produce novel research contributions that were not possible with the traditional slice-by-slice reviewing technique.

Their interactions with the remaining groups of actors is really vivid. They participate in the software development process as well as in the research stage where they provide visualization researchers with suitable data for testing. They form a very unique group in the software development process of medical visualization software as they hold very strong connections to all other actors involved in the medical visualization software development process.

4. Requirements for Medical Visualization Software

As Section 3 showed, developing medical visualization software requires different actors. Developing medical visualization itself comes with a set of requirements that we aim to summarize in this Section. Considering the different actors, they all apply a different importance to the existing requirements, that we aim to differentiate. Interestingly, a full set of requirements in medical visualization software development was not formulated so far. Gillmann et al. [GLWH16] formulated requirements that need to be fulfilled in order to create a visualization that is suitable for applications. We use this list as a basis and extended it with requirements identified throughout representative from all groups of actors involved in the medical visualization software development. Here, we tried to evaluate their personal impression of the importance of each requirement while fulfilling their daily tasks. They were allowed to express the importance of each requirement from 1 (low) to 5 (high). In addition we aim to indicate the importance of each actor in the medical visualization development process to meet this requirement, as shown in Figure 3.

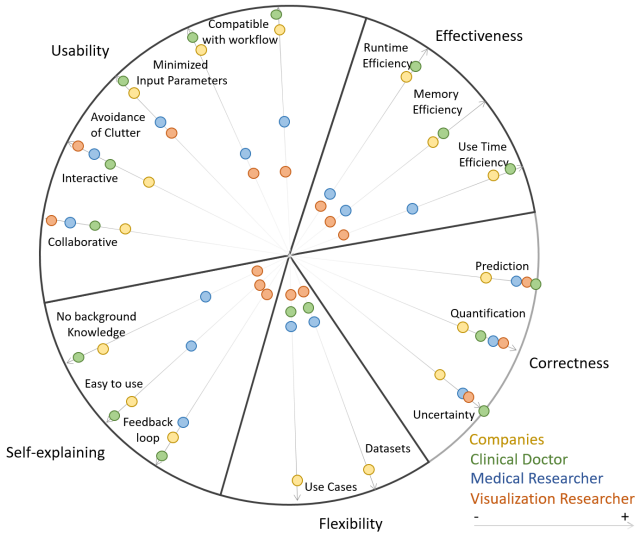

Figure 3: *Requirements for successful visualization in applications based on Gillmann et al. [GLWH16]. The importance of each actor in medical visualization development is visualized by the distance to the center in the given visualization. The closer a point is to the center, the less important the requirement is. The closer the point is to the boundary of the circle, the more important the respective requirement is.*

The requirements are divided into 5 groups: usability, effectiveness, self-explaining, flexibility and correctness as shown in Figure 3.

Considering correctness, a medical software needs to assist medical doctors in the prediction and quantification of diseases. This process is usually affected by uncertainty [WG84], which needs to be communicated in the medical visualization software. Here, all four actors aim for high correctness.

Flexibility contains the requirements to support different use cases as well as different input datasets. The importance of these requirements differ depending on the selected actor. Medical researchers and clinical doctors are often highly specialized in one specific field in medicine. The medical visualization software needs to provide them with a tool that helps them solve their daily tasks independent of the tool and its ability to solve further tasks. As a

company, it is important to offer a wide range of products that can be sold to many clinics. Here, many use cases and datasets are provided to obtain a large portfolio. Visualization researchers aim to develop novel visualization prototypes for given datasets arising in their collaborations. Depending on the other actor in the collaboration, the importance of flexibility may vary to them, but in general the flexibility is not a driving force in their prototyping process.

Self-explaining is an important feature, especially in the medical area. Here, clinicians require a visualization that does not require background knowledge in visualization techniques. An easy to use tool with a feedback loop to make use of the visualization should be given. Clinicians normally have minutes to make decisions and therefore self-explaining is an important feature. This directly transfers to companies when they aim to develop a sellable product. Medical researchers on the other hand require a feedback loop in the visualization as well in order to understand their results, but they do not require a super easy to use tool that can be used without background knowledge. During their research they aim to make new findings, not making them as quick as possible. In addition, they are more open for novel visualization approaches if it benefits their scientific outcome. Visualization researchers aim to develop self-explainable visualizations, but for them this requirement is not as important as they are experts in the field.

Effectiveness regarding runtime, memory and use-time efficiency is important for clinicians. They need to make decisions fast. Companies try to fulfill the needs of clinicians and aim to offer effective visualization software. On the other hand, visualization researchers do not extensively care about efficiency. The prototypes they develop are required to show their scientific contribution and are not developed to be used in clinical daily routine directly. Medical researchers have similar requirements here, but focus a bit more in the use-time efficiency as they try to make new findings.

At last, usability unions collaborative, interactive visualization techniques that avoid clutter. They can be used with minimal input parameters and are compatible with the current workflow in medicine. These requirements are highly important to medical researchers and so they are for companies. On the other hand, interactivity and collaborativity are nice to have for clinicians, but a solution that is not depending on this feature is preferable in order to be fast to use. Visualization researchers on the other hand aim to provide interactive and collaborative systems, as they have learned their importance throughout their education. Medical researchers appreciate those features as it gives them the opportunity to explore and understand their data better.

In general it can be observed, that we obtain two subgroups of actors that are connected to each other in terms of the made requirements. Clinicians and companies as well as visualization researchers and medical researchers. Between these two subgroups exists a gap driven by the different importance of requirements in medical visualization development and the legal frame that restricts the medical software development process.

5. Proposed Workflow

According to the legal frame in medical visualization development and considering the different actors as well as their focus on diverse requirements, we aim to specify and adapt an agile variant of the V-Model. This choice is motivated by the international standart IEC 62304 [Int06], which promotes use of the V-Model [FM91]. Therefore the V-Model is often utilized for medical software development as it was accepted , through IEC 62304, as a harmonized standard by both the European Union and the United States. This might also be the case for the MDR, if this standard will be accepted as a harmonized standard under the MDR.

The agile variant of the V-Model will have more active actors, corresponding to the actors defined in Section 3. Therefore, we differentiate two different groups of interests that have been detected in Section 4 and provide a mechanism that connects their development processes. The general idea is to split several phases of the Model in order to improve and accelerate the development process [Com].

5.1. The V-Model

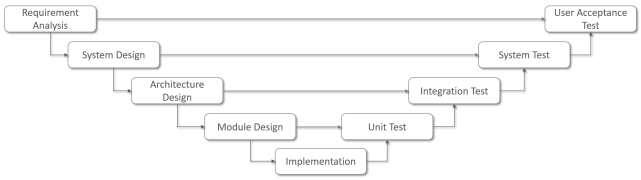

Figure 4: *The original V-Model containing Requirement Analysis, System Design, Architecture Design, Module design and Implementation on the left side. The phases are opposite to the Unit Tests, Integration Test, System Test and User Acceptance Test that are shown on the right side.*

The V-Model is a non-linear extension of the waterfall model [Ben83], building relationships between each stage of the development life cycle and the testing phases. It provides a design hierarchy, that separates the software development process into different tasks.

First, the Requirement Analysis declares user needs and formulates requirements for an ideal system as seen by the user. These needs and requirements are reviewed for feasibility in the System Design phase, creating a general model of the system, providing appearance samples of system components. The Architecture Design phase describes all modules, their functionalities and relationships between the modules. The Module Design defines modules which are broken down into manageable units, describing details of the implementation such that programming the application can start immediately .

Until now the project is being planned and no actual programming was carried out. The verification/planning phase is complete at this point and the implementation phase begins. Subsequently, each planning stage is being checked against its deployed requirements in the validation/testing phase. The design of the tests were already defined in the corresponding planning phase, such that the tests can be implemented and executed.

The first stage of testing, Unit Tests, regards the smallest entities called units which are tested for functionality when being isolated from other units. The Integration Test then ensures that these

units can communicate with each other. The whole application is tested in the following System Test, making sure that all parts of the application are working as expected and that the performance is adequate. Hereinafter the User Acceptance Test ensures that the software can work in a production environment, utilizing real world datasets, assuring real time functionality.

5.2. Agile V-Model

The V-Model seems to be a very rigid scheme at first, unable to adapt to changes during development. But this is not the case, as the V-Model does not describe the order of the single stages, but their dependencies. Therefore, the Requirement Analysis stage does not have to be complete to begin the System Design.

This way, single parts of the software can be analysed, designed and implemented, while other parts are not even planned. Various modules can be in different stages of development. The Requirement Analysis as well as the Software design can be adjusted to match the overall direction of the project when more and more parts are being finished and tested. This allows for great flexibility and enables the usage of agile Software development schemes. Examples for the implementation of the agile software development using the V-Model are already available [ÖTM17, Ble14, Com].

5.3. Adaptions to the V-Model

Figure 1 shows the adapted V-Model that captures all actors and highlights who is in charge of which phase. In contrast to the original V-Model in Figure 4, we propose to split the original left half of the steps (Requirement Analysis, System Design, Architecture Design, Module design). The goal is to create a space for visualization researchers and medical researchers to build prototypes and directly pass their findings into the original V-Model that is processed by a company to create a final product.

In the medical visualization prototype phase, medical researchers and visualization researchers work together in order to create a first prototype that can be discussed with companies and clinicians. Here, the verification and implementation phases of the original V-Model are conducted. Requirement Analysis and System Design will be defined by the medical researcher that clearly needs to express their requirements for the software to develop. In conjunction to visualization researchers, medical researchers develop an Architectural Design for the visualization software prototype. Followed by that, visualization researchers develop a Module design and implement a prototype of the medical visualization software, that will be tested quickly by the medical researchers. Here, a new phase is added that summarizes the lessons learned in the prototyping process and giving important knowledge into the actual software development process.

After each phase in the prototyping process, made conclusions and found dependencies need to be collected and passed to the corresponding phase in the medical visualization software development process. Here, companies as well as clinicians can benefit from the experiences in the prototyping process. When clinicians formulate their Requirement Analysis and System Design, they can build them on top of the formulated requirements in the prototype phase. This saves time and minimizes the risk of missing important requirements. These benefits continue during the Architecture Design, Module Design and Implementation of the medical visualization development phase. Findings from the prototyping process helps clinicians and companies to process remaining phases faster with better prior knowledge.

After the implementation of the medical visualization software is achieved, the V-Model requires a set of tests, proofing that the developed software does what it is actually supposed to do. These phases will be highly influenced by the lesson learned phase from the prototyping phase. In this phase, medical researchers and visualization researchers formulate benchmarks, that the medical software needs to fulfill. This will refine the testing phases and helps identifying miss-engineered aspects in the medical visualization software.

When testing the medical visualization software, Unit Tests are up to the company. For the remaining tests, a close collaboration of all actors in the medical software development process is required. Medical researchers and visualization researchers summarized their experiences when developing the prototype of the medical software. This gives important hints on how to test the medical software and which benchmarks need to be achieved. On the other hand, clinicians further restrict the behavior of medical software in terms of clinical applicability. Here, companies need to address these requirements in order to produce an applicable medical visualization software.

The testing phase brings to light flaws of the planning phase, as ideas and decisions made in the planning phase can now now be tested for feasibility. Some parts of the system will not function the way they were intended to, others can have unwanted side effects. To account for this, information from the tests can also influence the planning phase to make improvements. These improvements in the planning phase will then also change the way of testing the application.

6. Open Problems

Although we presented an approach that helps enhancing and accelerating the software development process of medical visualization software, the legal frame and the differing interests of actors are not the only factors that prohibit the successful transfer of visualization research into clinical application. We investigated further problems in this process, that will be summarized in this section.

6.1. Transparency of visualization/computational approach

When clinicians consult medical visualization software to fulfill an arbitrary task, transparency is an absolute must [DGC08]. No matter how accurate or reliable a software seems to be, clinicians are responsible for their decisions and they cannot blame a wrong decision on a used software. Resulting from this, a medical visualization can only be used when clinicians are able to understand the process of computing conducted by the software. More precise, a medical visualization software needs to assist a clinician, not replace him. Here, medical visualization software needs to be more transparent in order to create trust of the using clinician.

6.2. Education/Information of novel visualization techniques

Many clinicians are not aware of novel visualization techniques and even if they are, it takes time to get used to a new software, no matter how intuitive it is. Here, clinicians often cannot invest the time to learn how to use new software, understanding the limits and benefits they get. Gillmann et al. [GWHH17] provided a concept to teach novel medical image processing and visualization tools to medical students. The use of such courses needs to be highly promoted in order to make clinicians and medical researchers aware of which products exist.

6.3. Platforms for improving the connection of actors

As we show in this manuscript, there exist different gaps between the actors of medical visualization software development. Although medical researchers and visualization researchers al well as companies and clinicians have a strong connection, but these links do not hold for each pair of actors. In many cases this is due to a lack of knowledge of possible members in different actors. Therefore, medical software development requires a suitable platform that brings together the different actors.

6.4. Software development tools for MDR proofed development

Many visualization approaches that are developed in visualization research lack the ability to be easy enough to adapt for companies. This results from multiple reasons, such as reproducibility of code from scientific papers or extremely prototyped code, if provided by the researcher. Here, a systematic approach for explaining and publishing medical visualization approaches is required in order to attract companies to implement them into their software. On the other hand, this approach should only add a minimal extra effort for visualization researchers as they are not paid to develop ready to use software.

6.5. Open Source Policies

In many cases the transformation of a visualization approach from the prototype development style into the company development style can be a hard step to overcome. Here, companies should provide open source development environments that can be used by visualization researchers. This allows visualization researchers to directly develop in the companies development style and minimizes the effort to transfer visualization techniques into a development environment. If companies are afraid of loosing control over their achievements, they can remove features from their open source software.

6.6. Approval of IEC 62304 as a harmonized standard of the MDR

As we proposed in this paper, a suitable workflow that includes all actors of the software development process is required. We can only propose this workflow, but an application needs to be dictated by the MDR. As a starting point, the IEC 62304 should be defined as a harmonized standard of MDR [Eur] in order to achieve uniqueness in the software development process of medical visualization software. As an additional step, the workflow proposed in this paper should at least be suggested by MDR.

6.7. Lack of Up-To-Date Hardware in Medical Environment

Contrary to computer science where large efforts are made to maintain an up-to-date Hardware environment, the medical sectors often lacks the financial support to keep their hardware up-to-date. This is based on other priorities, e.g. buying new medical devices or hiring more staff members. Resulting from this, novel medical software might not be able to be utilized although clinicians wish to. Here, financial sources should be generated that are reserved for maintaining an up-to-date hardware setup in clinics.

7. Conclusion

In this paper we determined a clear gap between medical visualization research and clinical applicability, giving suggestions on how to close this gap. We summarized the status quo in medical visualization and analyzed the actors in medical visualization development and their requirements in this process. Based on this, we propose a novel workflow in medical visualization research including all actors.

References

[AA19] ABDALLAH Y. M. Y., ALQAHTANI T.: Research in medical imaging using image processing techniques. In *Medical Imaging-Principles and Applications*. IntechOpen, 2019. 5

[Agf20] AGFA-GEVAERT GROUP: Tiphce, 2020. https://www.tiphce.com/. 2

[Ben83] BENINGTON H. D.: Production of large computer programs. *Annals of the History of Computing 5*, 4 (1983), 350–361. 6

[Bis18] BISWAS M.: State-of-the-art review on deep learning in medical imaging. *Frontiers in Bioscience 24* (10 2018). doi:10.2741/4725. 2

[Ble14] BLESS M.: Scrum und die IEC 62304. *http://www.scrum-und-die-iec62304.de* (2014). 7

[BPK*12] BOTHA C., PREIM B., KAUFMAN A., TAKAHASHI S., YNNERMAN A.: From individual to population: Challenges in medical visualization. *Mathematics and Visualization 37* (06 2012). doi:10.1007/978-1-4471-6497-5_23. 2

[BSV*18] BOER L., SCHAAF M., VINCKEN K., MOL C., STUIJFZAND B., VAN DER GIJP A.: Volumetric image interpretation in radiology: scroll behavior and cognitive processes. *Advances in Health Sciences Education 23* (05 2018). doi:10.1007/s10459-018-9828-z. 1

[Cap] CAPTERRA INC.: Capterra. www.capterra.com/medical-imaging-software. Accessed: 2020-03-06. 2

[Chi01] CHITTARO L.: Information visualization and its application to medicine. *Artificial intelligence in medicine 22*, 2 (2001), 81–88. 1

[Cla] CLARIVATE ANALYTICS: Web of science. webofknowledge.com. Accessed: 2020-03-06. 1, 2

[Com] COMTRADE: Agile - a better choice in the development of software for medical devices. https://content.comtradedigital.com/agile-softwareentwicklung-medizintechnik. Accessed: 2020-03-09. 6, 7

[Cou17a] COUNCIL OF EUROPEAN UNION: Council directive 80/181/eec of 20 december 1979 on the approximation of the laws of the member states relating to units of measurement and on the repeal of directive 71/354/eec, 2017. https://eur-lex.europa.eu/eli/dir/1980/181/oj. 4

[Cou17b] COUNCIL OF EUROPEAN UNION: Regulation (eu) 2017/745 of the european parliament and of the council of 5 april 2017 on medical devices, amending directive 2001/83/ec, regulation (ec) no 178/2002 and regulation (ec) no 1223/2009 and repealing council directives 90/385/eec and 93/42/eec, 2017. https://eur-lex.europa.eu/eli/reg/2017/745/oj. 1, 3

[Cou17c] COUNCIL OF EUROPEAN UNION: Regulation (eu) 2017/746 of the european parliament and of the council of 5 april 2017 on in vitro diagnostic medical devices and repealing directive 98/79/ec and commission decision 2010/227/eu, 2017. https://eur-lex.europa.eu/eli/reg/2017/746/oj. 3

[DGC08] DONG F., GHINEA G., CHEN S. Y.: *User Centered Design for Medical Visualization (Premier Reference Source)*, 1 ed. IGI Global, USA, 2008. 7

[Eur] EUROPEAN COMMISSION: Harmonised standards generated by the has (harmonised standards) database. https://ec.europa.eu/growth/single-market/european-standards/harmonised-standards/medical-devices_en. Accessed: 2020-03-10. 8

[FM91] FORSBERG K., MOOZ H.: The relationship of system engineering to the project cycle. In *INCOSE International Symposium* (1991), vol. 1, Wiley Online Library, pp. 57–65. 6

[GAH*17] GILLMANN C., ARBELÁEZ P., HERNÁNDEZ J. T., HAGEN H., WISCHGOLL T.: Intuitive Error Space Exploration of Medical Image Data in Clinical Daily Routine. In *EG/VGTC Conference on Visualization (EuroVis) - Short Papers* (2017). doi: 10.2312/eurovisshort.20171148. 2

[GE 20] GE HEALTHCARE: Ge advanced visualization, 2020. https://www.gehealthcare.de/products/advanced-visualization. 2

[GH02] GARVEY C., HANLON R.: Computed tomography in clinical practice. *BMJ (Clinical research ed.) 324* (06 2002), 1077–80. doi:10.1136/bmj.324.7345.1077. 2

[GLWH16] GILLMANN C., LEITTE H., WISCHGOLL T., HAGEN H.: From Theory to Usage: Requirements for successful Visualizations in Applications. In *IEEE Visualization Conference (VIS) - C4PGV Workshop* (2016). 5

[GS14] GILLMANN C., SALZ P.: Improving visual communication for eit-based lung research. In *2014 IEEE Pacific Visualization Symposium* (2014), pp. 291–295. 2

[GWHH17] GILLMANN C., WISCHGOLL T., HERNÁNDEZ J. T., HAGEN H.: Teaching Image Processing and Visualization Principles to Medicine Students. In *To be published in IEEE Visualization Conference (VIS) - Pedagogy of Data Visualization Workshop* (2017). 8

[Int06] INTERNATIONAL ELECTROTECHNICAL COMMISSION: International iec standard 62304 first edition 2006-05, 2006. 6

[Int18] INTRASENSE: Intrasense, 2018. http://www.intrasense.fr/. 2

[Kon04] KONINKLIJKE PHILIPS N.V.: Phillips healthcare, 2004. https://www.philips.de/healthcare/solutions/klinische-it/. 2

[Kon20] KONINKLIJKE PHILIPS N.V.: Intellispace portal 11, 2020. https://www.philips.de/healthcare/product/HC881103/intellispace-portal11/uberblick. 2

[LSBP18] LAWONN K., SMIT N. N., BÜHLER K., PREIM B.: A survey on multimodal medical data visualization. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 413–438. 2

[ÖTM17] ÖZCAN-TOP Ö., MCCAFFERY F.: How does scrum conform to the regulatory requirements defined in mdevspice®? In *International Conference on Software Process Improvement and Capability Determination* (2017), Springer, pp. 257–268. 7

[ROJP11] ROPINSKI T., OELTZE-JAFRA S., PREIM B.: Visual computing in biology and medicine: Survey of glyph-based visualization techniques for spatial multivariate medical data. *Computers and Graphics 35* (04 2011), 392–401. doi:10.1016/j.cag.2011.01.011. 2

[Ter20] TERARECON: Terarecon products, 2016-2020. https://www.terarecon.com/. 2

[WG84] WILSON R., GRANLUND G. H.: The uncertainty principle in image processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6*, 6 (Nov 1984), 758–767. doi:10.1109/TPAMI.1984.4767599. 5