

A Hybrid Assessment Approach for Medical Device Software Development Companies

Özden Özcan-Top, Fergal McCaffery

Regulated Software Research Centre & Lero
{ozden.ozcantop; fergal.mccaffery}@dkit.ie
Dundalk Institute of Technology
Dundalk, Ireland
+353 429 37 0462

Abstract: Medical device software development organisations are bound by regulatory requirements and constraints to ensure that developed medical devices will not harm patients. Medical devices have to be treated as complete systems and be evaluated in this manner. Instead of manufacturers having to ensure compliance to various regulatory standards individually, the authors previously developed a medical device software process assessment framework called MDevSPICE[®] that integrates the regulatory requirements from all the relevant medical device software standards. MDevSPICE[®] was developed in a manner that suits plan-driven software development. In order to improve the usability of MDevSPICE in agile settings, we extended the assessment approach. The hybrid assessment approach described here, combines the MDevSPICE[®] based process assessment method with steps for prioritization of improvement needs through value stream mapping, and enabling process improvement through the use of KATA. This approach integrates agile methods into the medical device software development process whilst adhering to the requirements of the regulatory standards. This paper describes the implementation of the approach within four organisations that develop software in line with medical device regulations.

Keywords: MDevSPICE, medical device software development, medical standards, agile software development

1. INTRODUCTION

Increasingly, safety-critical software systems are affecting our everyday living as more and more software is embedded into safety critical systems such as cars, trains, airplanes, and healthcare systems which include medical devices. In order to ensure the safety of such systems international standards have been developed. In order to market a medical device within a particular region, the manufacturer has to satisfy a number of regional regulatory requirements. To assist software development organisations in the medical device domain and to achieve

regulatory compliance, the authors have previously developed an integrated framework of medical device software development best practices called MDevSPICE® [1]. MDevSPICE® integrates generic software development best practices with medical device standards' requirements enabling robust software process assessments to be performed against an organisation's current software development practices.

MDevSPICE® was originally designed to be used with both traditional waterfall and V-model lifecycle models. These software development life cycle models are plan-driven, rigid and sequenced which make MDevSPICE®'s usage limited for companies which wish to adopt agile software development approaches. In order to extend the capability and usage of MDevSPICE®, agile practices were integrated into the assessment approach.

A standard MDevSPICE® assessment approach includes evaluation of 23 processes at the base practice level. In this paper, we describe the hybrid assessment approach, which includes value stream mappings. Additionally, it integrates agile practices into the reporting phase of an MDevSPICE assessment so that such practices can be provided to organisations within the findings report instead of purely plan driven improvement recommendations. Afterwards, we prioritise these recommendations to enable a focus to be provided to the process improvement implementation through the use of KATA [2]. We discuss how the hybrid assessment approach was implemented in four companies, three of which are from medical domain and one of them is a software development company that is planning to develop software that could in the future be classified as a medical device.

In Section 2, we provide an overview of the regulatory requirements medical device software development companies face before they are able to market their devices. In Section 3, we describe the structure of the MDevSPICE® framework. Section 4 outlines the challenges associated with using MDevSPICE® lessons learned when validating the framework in expert reviews and in industry through MDevSPICE® pilot assessments. In Section 5, we present the

literature review and how the perception of the use of agile software development in medical device software development has changed. In Section 6, we introduce the hybrid assessment approach for medical device software development companies. In Section 7, we present the implementation of the approach in four companies. We provide our conclusions and future research in Section 8.

2. REGULATIONS

In order to place a medical device on the market it is necessary to satisfy the regulatory requirements for that region. Two significant markets for medical devices are the USA and the European Union. Within the USA the FDA are responsible for issuing and managing medical device regulation. The FDA provides a number of regulatory and guidance documents in relation to medical devices, including the Code of Federal Regulation Title 21, Chapter I, Subchapter H, Part 820 [3]. In the European Union (EU), the European Commission defines directives that are then overseen by competent authorities within each of the member states. There are a number of medical device directives: the Active Implantable MDD (AIMDD) 90/385/EEC [4]; the general Medical Device Directive (MDD) 93/42/EEC [5]; the In-vitro Diagnostic (IVD) MDD 98/79/EC [6]; and an amendment which applies to all three directives 2007/47/EC [7].

IEC 62304:2006 [8] is the foundational medical device software standard. This standard describes medical device software lifecycle processes and outlines software safety classification. IEC 62304 is an important standard for medical device software developers as it is the only standard that provides information for medical device software based on the worst case scenario of software failure. As risk management is important for medical device software development IEC 62304 should be used along with the general medical device risk management standard ISO 14971 [9] and IEC 80002-1 [10]. IEC 80002-1 provides guidance on the application of ISO 14971 for software development. Additionally, within the medical device

domain it is important to have a Quality Management System (QMS) in place. For example, the QMS for the European medical device domain is ISO 13485 [11]. The QMS requirements of ISO 13485 need to be adhered to in order to market a medical device in Europe. As ISO 13485 does not specifically focus upon software requirements, it may be used to support the safe design of medical device software in conjunction with IEC 62304 [8].

IEC 62304 defines three medical device software safety classifications: Class A; Class B; and Class C. Failure of software that is of safety Class A will result in no injury or damage to the health of a patient. In the case of failure of software with a safety classification of B, non-serious injury may occur. Software with a safety Classification of C has the highest risk as in the case of failure death or serious injury may occur happen. Depending on the functionality of software within the medical device, the software safety classification may vary from the overall medical device safety class.

IEC 62304 considers a medical device to include a software component as rather than being a complete medical device system. Consequently, IEC 62304 does not consider a medical device to be formed completely of software, but rather that there are also other electronics or hardware components that together with the software form a complete medical device system. Therefore, system level requirements are not included within IEC 62304 but instead within IEC 60601-1 [12] which is a medical device system level standard. Additionally, given the increasing importance of human factors and usability within the medical device industry, organizations should also adopt IEC 62366 [13], which outlines usability engineering requirements for the medical devices.

As a result of an amendment to the medical device directives [7] software systems that perform the functionality of a medical device (as opposed to being a component of an overall medical device) could be defined as a medical device in their own right. However, until recently international medical device standards did not address standalone software as a medical device.

To address this IEC 82304-1 [14] was published in October 2016. IEC 82304-1 applies to healthcare software that is designed to operate on IT platforms without dedicated hardware, e.g., mobile applications on tablets and phones.

If a company is planning to market a medical device in the US, they need to register their product with the FDA. The FDA have issued an overview of their guidance documents for developers of medical device software [15]. There are four guidance documents: the FDA Guidance on Premarket Submissions [16]; the FDA Guidance on Off-The-Shelf Software Use in Medical Devices; the FDA General Principles of Software Validation [17]; and Guidance on MMAs [18] to provide clarity for MMA developers in terms of what types of applications will be regulated and how. In order to market a medical device in Europe, organizations need to comply with the following directives: the Active Implantable MDD; the general Medical Device Directive; the In-vitro Diagnostic MDD; and 2007/47/EC.

3. MDevSPICE® FRAMEWORK

As outlined in the previous section organisations developing medical device software must adhere to a large number of regulatory requirements and international standards if they wish to market their device. In order to reduce the demanding and costly overhead associated with preparing for regulatory audits, we developed MDevSPICE® [1, 19]. MDevSPICE® integrates software requirements from the various international medical device standards and guidance documents (described in Section 2) with best practices for software and systems development into a single reference source. The capability of an organization's medical device software processes may then be assessed using this reference source. MDevSPICE® was designed to assist organisations to develop medical device software through adopting either a traditional waterfall model or a V-model lifecycle model. However, such development lifecycles are very

rigid, prescribed and sequenced, which presents problems in relation to introducing requirements changes throughout development.

As stated previously, IEC 62304 only relates to software that is a component of an overall medical device. Therefore, IEC 62304 only includes software lifecycle processes, and additional processes are required when the software developed is the complete medical device system. Consequently, when software is the entire medical device systems development processes also need to be considered. These systems development processes were derived from ISO/IEC 12207:2008 [20]. Additionally, Risk Management and Quality Management requirements from ISO 14971 and ISO 13485 were also added to MDevSPICE® as both are foundational standards for medical device development. MDevSPICE® consists of 23 processes [1]. There are ten system lifecycle processes, 8 software lifecycle processes and 5 support processes. MDevSPICE® has been validated through both international expert review and industrial trials and was founded upon IEC TR 80002-3, which is the Process Reference Model for IEC 62304 [21]. The development of IEC TR 80002-3 was led by the authors in collaboration with the international medical device standards working group for IEC 62304. Upon successful completion of international expert reviews, MDevSPICE® was validated through pilot assessments in 10 medical device software development organizations over two years.

MDevSPICE® was designed to enable full lifecycle coverage in a plan-driven manner. It provides the requirements of all associated medical device standards in one place. This framework integrates software engineering best practices so that safer medical device software may be developed through adopting MDevSPICE®.

4. The CHALLENGES ASSOCIATED with USING MDevSPICE®

Formal models like CMMI [22] and ISO/IEC 15504 [23] are used to promote maturity and capability in software processes. In the literature, there are very few studies discussing the challenges and successful implementations of integrating agile and formal software process improvement models [24-26]. Literature discussions are mostly shaped to answer the question of why agile and formal process improvement models do not contradict each other. Previously, it was thought by the commercial software development community that formal process improvement models and agile are at odds with each other [27, 28]. According to the Software Engineering Institute (SEI) what caused the perception of the relationship between Agile and CMMI to be like “oil and water” may be attributed to the presentation of extreme examples of both sides by early adopters, therefore providing inaccurate information about them and their misusages [27].

In a study including 40 agile experts, composed of researchers and academics, the participants classified each CMMI component as either: (1) in conflict with; (2) of no consequence to; or (3) being supportive of agile methods in general. The results showed that the differences between the agile and formal process communities are beginning to be better articulated and better understood. It was stated that although there are significant differences between both sides, they could work together [29].

The authors have performed many MDevSPICE® assessments within medical device software development companies and have experienced challenges using MDevSPICE® in Agile Software Development environments. MDevSPICE® is designed for high-risk, safety critical systems which in the case of a software failure, could result in harm to human life. To reduce the risk associated with software failures, MDevSPICE® favors upfront planning. This is one of the major contradictions of MDevSPICE® with agile software development that agile methods favor an adaptive and exploratory approach through accepting that it is difficult to get requirements completely understood up-front. For example, signing off both system and

software requirements before development commences is in conflict with the product backlog grooming practice in Scrum which allows you to dynamically update and re-prioritize backlog items.

Another challenge that the authors have observed was associated with the recommendations given to organisations after process assessment. As a descriptive model, MDevSPICE® does not prescribe how to perform base practices within a specific context, however provides guidance with possible outputs of the practices. These outcomes again, are derived from formal software development life cycle models with recommendations being plan driven and formal. Therefore, more effective ways to implement software needed to be integrated with MDevSPICE®.

Additionally, companies found that MDevSPICE® was excellent at highlighting gaps in current software development processes but that the plan-driven recommendations provided often increased the overhead associated with development. Therefore, they really would like to have been provided with more flexible recommendations that could be implemented in a much more efficient manner. Likewise, companies also found that MDevSPICE® tended to focus more upon providing guidance in relation to strategies and plans that need to be put in place rather than providing more practical recommendations as to how this could be achieved.

5. AGILE USAGE in the MEDICAL DOMAIN

This section has been taken out of the pre-print version due to copy right issues. The full paper could be found at <https://onlinelibrary.wiley.com/doi/full/10.1002/smr.1929>.

6. The HYBRID APPROACH

The hybrid assessment approach we discuss within this paper for medical device software development has two principal activities as shown in Figure 1: studies in *theory* and studies in *the field*. These two studies are fed from each other. In theory, we are developing an integrated

agile MDevSPICE[®], by analyzing 1st and 2nd generation agile software development approaches at a practice level and mapping these practices with MDevSPICE[®] base and specific practices. Thus, we ensure both agility and conformance to regulatory standards. The studies *in the field* start with basic MDevSPICE[®] based assessments to identify strengths and issues within the companies. The basic MDevSPICE[®] assessment process includes assessment planning, execution and reporting phases. This process extends the MDevSPICE[®] assessment process by including both prioritization and implementation of the prioritized recommendations. A value stream mapping is performed as part of the prioritization stage to observe where the bottlenecks are between the process blocks. The prioritized recommendations are implemented through the use of the KATA technique [49] in an iterative manner until all recommendations are complete. The following sections provide details of these activities.

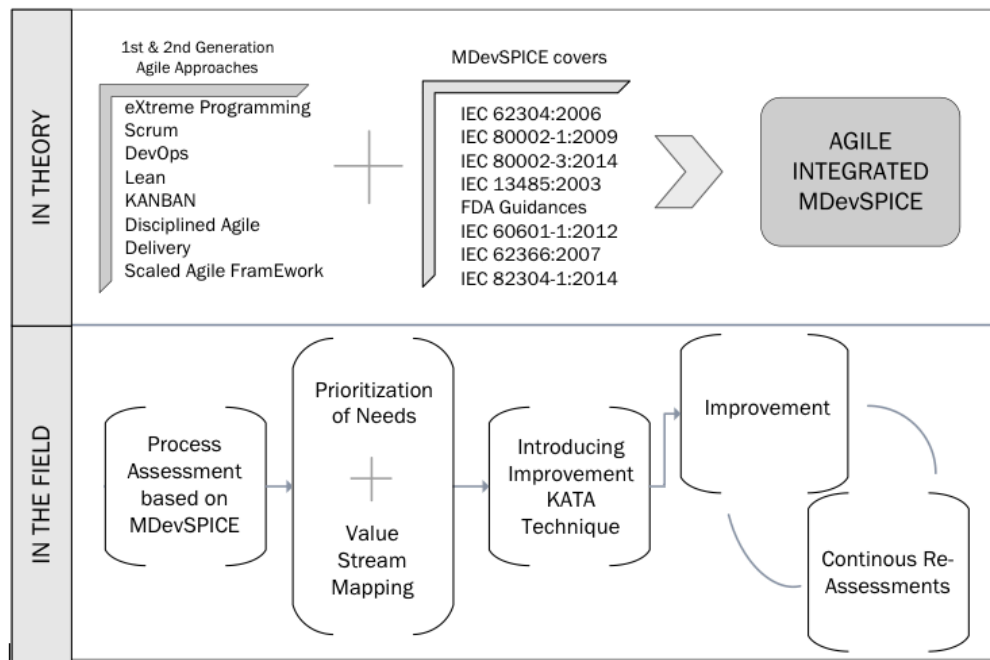


Figure 1 Representation of the Hybrid Assessment Approach

6.1. Studies in Theory

One of the major components of our hybrid assessment approach is the “agile practices repository” that includes a bilateral mapping of agile practices against MDevSPICE[®]’s base and specific practices. The purpose of the mapping is to understand which of the MDevSPICE[®]

base and specific practices could be fulfilled by adopting associated agile practices (for recommendations) and what additionally needs to be done to satisfy regulatory requirements. We aim to extend the applicability of MDevSPICE® in companies which follow an agile software development life cycle. We selected eXtreme programming [50], Scrum [51], DevOps[52], Lean [53], Kanban [54], Disciplined Agile Delivery (DAD) [31] and Scaled Agile Framework (SAFE) [55] as the source of the agile practices that we adopt. This selection was made to cover: small and large scale agile adoption (DAD and SAFE); fundamental agile methods (XP, Scrum, KanBan); and new paradigms (DevOps, Lean). For example, in Table 1 we provide a mapping between the primary practices of XP and MDevSPICE®’s base practices.

Table 1 The Mapping between Primary Practices of XP and MDevSPICE®’s Processes and Base Practices

Primary Practices of XP	MDevSPICE® Process	MDevSPICE® Base Practice/s
Sit Together	PRO.1 Project Planning	PRO.1.BP6: Define needs for experience, knowledge and skills. PRO.1.BP10: Establish project plan.
Pairing and Personal Space	PRO.1 Project Planning	PRO.1.BP6: Define needs for experience, knowledge and skills.
Weekly Cycle	PRO.2 Project Assessment and Control	PRO.2.BP2: Monitor project interfaces PRO.2.BP3: Report progress of the project PRO.2.BP4: Perform project review
Quarterly Cycle	PRO.2 Project Assessment and Control	PRO.2.BP2: Monitor project interfaces PRO.2.BP3: Report progress of the project PRO.2.BP4: Perform project review
Whole Team	PRO.1 Project Planning	PRO.1.BP6: Define needs for experience, knowledge and skills.
Informative Workspace	Could be PRO.1 Project Planning	Not Exist
Energized Work	Could be PRO.1 Project Planning	Not Exist
Pair Programming	DEV.4 Software Unit Implementation and Verification	DEV.4.BP1: Implement the software units.
Slack	PRO1.Project Planning	PRO.1.BP4: Define and maintain estimates for project attributes. PRO.1.BP8: Define project schedule.
Ten Minute Build	DEV.5 Software Integration and Integration Testing	DEV.5.BP1: Integrate software units into software items. DEV.5.BP2: Verify that software integration follows integration strategy. DEV.5.BP3: Develop tests for integrated software items.

Continuous Integration	DEV.4 Software Unit Implementation and Verification DEV.5 Software Integration and Integration Testing	DEV.4.BP4: Verify software units. DEV.5.BP1: Integrate software units into software items.
Test First Programming/Continuous Testing	DEV.4 Software Unit Implementation and Verification DEV.5 Software Integration and Integration Testing	DEV.4.BP4: Verify software units. DEV.5.BP3: Develop tests for integrated software items. DEV.5.BP4: Test integrated software items in accordance with the integration plan and document the results.
Incremental Design	ENG.2 System Requirements Analysis DEV.1 Software Requirements Analysis	ENG.2.BP1: Establish system requirements. ENG.2.BP5: Evaluate and update system requirements. DEV.1.BP1: Define and document all software requirements.
Story	DEV.1 Software Requirements Analysis	DEV.1.BP1: Define and document all software requirements.

6.2. Studies in the Field

MDevSPICE® Based Process Assessment: MDevSPICE®, has been validated in the medical device industry through pilot assessments during the past four years [1, 19]. An MDevSPICE® assessment is performed to grasp the current condition of the company and identify problems and issues regarding the medical software development life cycle. The basic MDevSPICE® assessment process includes assessment planning, execution and reporting phases as described below. In the **assessment planning phase**, we decide upon the scope of the assessment. In other words which project/s to assess and the processes to be included in the assessment. Additionally, the logistics are agreed in terms of the dates that the assessment will take place on and the company staff that will participate within the assessment.

In the **execution phase**, we perform interviews with the participants that were nominated for each process. One interview is performed for each process area with normally two assessors and at least one person from the company being present within each interview. During each interview, the assessors ask the company employees a comprehensive set of scripted questions that were derived from the base practices of each process by the assessment team in advance of

the onsite assessment. The assessors also ask additional related questions that arise during the course of the interview to glean more information about the companies' processes. In addition to the question and answer interview session, as we are assessing medical device organisations who have to produce evidence that they have followed their processes throughout the development lifecycle, we also request them to provide us with specific evidence of their software development process such as requirements documents, design artifacts and tools that are in actual use to support the responses to their interview questions.

In the **reporting phase**, upon completion of the onsite execution stage, the assessment team go offsite and prepare a findings report by working together to agree upon the assessment findings including strengths, issues and recommendations. The recommendations both include agile and plan driven practices. The assessment team then return back to the assessed company and present the findings report to all participants and this acts as the first step towards improvement. The next is to prioritize the recommendations provided in the findings report.

*A) **Prioritization:*** After the reporting phase, we obtain a list of the issues and problems that needs to be solved to deliver products in a more efficient manner. We prefer to keep these items on an Improvement Backlog which are prioritized and dynamic like a product backlog.

Value Stream Mapping: The reporting phase is followed by the value stream mapping study (VSM). The purpose of the value stream mapping is to visualize bottlenecks, redundancies and gaps in the value delivery system [56]. Therefore, it is very beneficial to perform this at the prioritization stage so that crucial bottlenecks can be highlighted and recommendations selected first that will enable the process to be improved in the most efficient order to remove these bottlenecks. VSM is one of the lean practices used in the Toyota Production System and was introduced by Rother and Shook [57]. The value stream is the flow of work from a customer request to the fulfillment of that request [49]. The value stream mapping is the process of identifying various process blocks at a decent level of detail within the product delivery. We

utilize the value stream of a product or feature to specify where to direct our attention for process improvement. To run a value stream exercise, we interview people with responsibilities for different parts of the software development lifecycle who can authorize the process changes required to achieve an effective future-state value stream. We use three metrics within the value stream mapping study as shown in Table 2: *Lead Time (LT)*, *Process Time (PT)* and *Percent complete and accurate (%C/A)* [49].

Table 2 Metrics that are used in Value Stream Mapping

Metric	Description
Lead Time (LT)	Total time to complete a process block with all interruptions
Process Time (PT)	Actual time to complete a process block without any interruptions
Percent Complete and Accurate (C/A %)	The proportion of time a process receives something from an upstream process that does not require rework

B) *Improvement KATA:* One of the first steps that must be taken in software process improvement is to decide how the changes will be identified, agreed upon and adopted in the existing system and more importantly how the improvement will be continuous after the assessment team (change agents) leave the company. The technique that we selected for this purpose is called Improvement KATA [2]. The KATA idea, which comes from Japanese martial arts, means habits or patterns of thinking that are being conducted by an individual and practiced every day [2]. Improvement KATA is a general-purpose framework including a set of practices for reaching goals where the path to the goal is uncertain. Improvement KATA provides an experimental approach with a 4 step pattern: 1) Understand the direction or challenge, 2) Grasp the current condition, 3) Establish the next target condition and 4) Iterate towards the target condition [49]. In process improvement, the experimental approach for finding the right practices and figuring out the ways to implement them is very important so that the companies do not over invest in a solution that will not work for them. The challenge of investing in the right improvement practices for the right time could only be achieved with the help of an experimental approach.

Following an MDevSPICE[®] based process assessment, we introduce companies to the Improvement KATA framework and the KATA document templates to be used in the process.

Some parts of this section have been taken out of the pre-print version due to copy right issues. The full paper could be found at

<https://onlinelibrary.wiley.com/doi/full/10.1002/smr.1929>

Improvement and Re-assessments: We implement the changes in an iterative and incremental way and follow the progress. Besides, to ensure the conformance to regulatory standards, we re-assess the software projects.

7. IMPLEMENTATION of the APPROACH via ACTION RESEARCH

We have implemented the Hybrid Assessment Approach which includes the following subsections described above: software process assessments based on MDevSPICE[®] (Section 6.2, Part (A)), prioritizations and value stream mappings (Section 6.2, Part (B)) and Improvement KATA approach (Section 6.2, Part (C)). Even though we have started improving the processes of the companies, the project is not in the re-assessment phase yet.

Based on the characteristics of this research, the best qualitative approach that suits the needs of the research is action research which associates theory and practice. In the action research, the practice is informed by the research and the research is informed by the practice synergistically [58]. “*Action research is an iterative process involving researchers and practitioners acting together on a particular cycle of activities, including problem diagnosis, action intervention, and reflective learning*” [58]. The purpose of implementing the proposed approach is to show the applicability of this approach in the medical device software development industry. Thus, we implemented the research in four different companies three of which are from medical device development domain and one of them is a software development organization that is planning to develop software that could in the future be classified as a

medical device and therefore also wants to implement MDevSPICE® as a process improvement framework.

7.1. Overview of the Companies and Planning

The selected companies approached us with the purpose of improving their software processes while ensuring regulatory requirements. After initial negotiations, they agreed to be a part of the study. In paragraph below, we briefly describe each participating company.

Company A develops medical applications for iOS, Android, Windows 8 and Web Browser. It was formed in 2011 and since 2012, it has been developing Medical Device Software. The products that they developed are classified as Class B based on IEC 62304:2006. It is a small company based in Ireland and including 7 people whom are developers, testers, a product manager and clinicians. *Company B* develops software that is currently not safety critical but the organisation has demands placed upon them from their industry as it has to be always accurate, reliable and consistent. Company B is based in Ireland and employs 50 people. *Company C* develops mobile and web applications to assist patients who are recovering from injury or operations or are dealing with chronic pain. The products that they develop are classified as Class B based on IEC 62304:2006. It is based in Ireland and 10 people work in the company. *Company D* develops medical device software applications for usage by patients, with an objective to improve patient engagement with healthcare practitioners. It is a large-scale company employing more than 150 people across three main offices in Ireland, Poland and the US. The medical device software products that they develop are mostly classified as Class B based on IEC 62304:2006.

In the planning phase, for each organization we have decided the type of the project and number of the processes that will take place in the process assessment, the dates of the assessment and the staff that will be included in the assessment.

Initially, we planned 1 full day of low level assessments for the major software life cycle processes with each company, 2 full days of off-site study for evaluation of the findings and report preparation and a further half day for the onsite findings' briefing presentation (for each company). We selected 10 processes to be assessed which were Project Planning, Project Monitoring, Stakeholder Requirements, System Requirements Analysis, System Architectural Design, Software Requirements Analysis, Software Architectural Design, Software Unit Implementation and Testing, Software Integration and Testing, Software System Testing. Prior to assessments, we defined a detailed set of assessment questions based on MDevSPICE® base practices. For value stream, prioritization and KATA studies, we prepared the templates.

7.2. Implementation of the Action Research within the Companies

We performed onsite visits to organisations for assessments. Two/three assessors were involved in each assessment, one asking scripted interview questions whilst the others took notes. In Table 3, we listed the roles of the interviewed staff, the total interview hours, types of assessed projects, and the assessed processes. The assessed processes diverged slightly from those planned for Companies A and B. Instead of the planned 10 processes, we actually assessed 7 for Company A and 6 for Company B due to the time required to ask the large number of assessment questions combined with observing the supporting evidence, as we operated within a fixed time for the assessment. The assessments within Companies A and B process were then followed by prioritization (including value stream mapping studies) and as planned. However, due to limited number of the assessed processes, we couldn't establish an overall view of the issues and problems in the projects within these assessments which had an significant impact on the prioritization phase.

Table 3 Key Figures of Action Research Conduct

	Company A	Company B	Company C	Company D
Interviewed Staff	Chief Technical Officer	Chief Technical Officer	Chief Technical Officer	Program Manager and

	and Chief Executive Officer		and Software Architect	Product Manager
Total interview hours	6 hours	6 hours	6 hours	11 hours
Assessed project type	Web and mobile based decision support tool	Tool to monitor individual and team performance in real-time	Mobile and web based exercise guidance software	A Platform which provides an infrastructure for a many medical device software applications which they also develop in-house.
Number of the Assessed Processes	7	6	23	23
Assessed Processes	<ul style="list-style-type: none"> - Project Planning - Project Monitoring - Stakeholder Requirements Definition - Software Architectural Design - Software Unit Implementation and Testing - Software Integration and Testing - Software System Testing 	<ul style="list-style-type: none"> - Project Planning - Project Monitoring - Stakeholder Requirements Definition - System Requirements Analysis - System Architectural Design - Software Requirements Analysis 	All MDevSPICE processes.	All MDevSPICE processes.

With the experiences obtained from the Company A and Company B process assessments, we consequently changed the assessment plan, and decided to perform a holistic assessment with 23 processes at a higher level (with less scripted questions). We reviewed all the assessment questions and defined 3 to 5 questions for each process which were more focused and open ended. For example, for system requirements analysis process, we asked the following four questions:

- What types of component/interfaces do you have?
- How do you define your system requirements?
- Do you make use of a traceability matrix?

- What challenges do you face in relation to the system requirements analysis process?

For each process, we repeated the last question given above for each of the 23 processes, in order to figure out the challenges that they faced.

In the value stream mapping studies, we discussed the major activities within the software development life cycle process, the lead times and percentage of rework required for these activities by the teams.

For Company A and Company B, the assessment team went offsite and worked as a team to prepare the assessment findings reports. Within the findings phase, in addition to identifying strengths and gaps, we also listed improvement action suggestions to address the issues. Due to time scheduling difficulties it took more than two months for the assessment team to meet with Company A and B again on their premises to present the findings presentation. In order to achieve this in a more effective way, we changed the plan and extended our onsite stay for Company C and Company D, and presented the high-level findings for all 23 processes within the same week.

The presentation of the findings and value stream mappings were followed by prioritization of the issues. We also presented how the improvement actions to resolves the identified issues could be implemented and presented the *Improvement KATA* approach to the development teams within each of the 4 companies. All 4 companies expressed excitement about the approach and liked the idea of implementing some initial actions to better determine if the suggested practices would work for them or not.

7.3. Findings of the Action Research

In this sub-section, we provide the findings of the action research in three parts: (a) common or specific issues that were found when performing the assessments, (b) an example of the

implementation of the value stream mapping study and (c) results of the prioritization of the issues.

(a) The Issues Found when Performing the Assessments

Some parts of this section have been taken out of the pre-print version due to copy right issues. The full paper could be found at

<https://onlinelibrary.wiley.com/doi/full/10.1002/smr.1929>.

Below, we have discussed the issues that were either common or specific to the four companies and our suggestions for resolving the issues. The issues listed here were identified through both conducting interviews and observing project artifacts during the onsite visits. We asked detailed, pre-defined questions during the onsite interviews for each *base practice* in MDevSPICE® to the process owners.

It was observed that Company A, B and C did not make use of formal project plans, project timelines or schedules. Progress against project objectives is not signed off. Knowledge is not transferred efficiently between projects. In terms of estimation, there is no size estimation of software but effort estimation is performed in an Ad-Hoc way.

We also found that customer involvement was quite limited within all the projects we assessed in the four companies. This creates accuracy and ambiguity problems in relation to captured requirements and as a result of this, rework was frequently required.

We discovered in all four companies that third party functionality and the risks associated with that functionality at the software requirements, system requirements and software architecture stages were not taken into account and were often overlooked. Special attention is needed to ensure that third party functionality would not introduce safety risks into the medical device.

In medical device development projects, most of the time hardware is part of the system and hardware decisions need to be made at the beginning of the projects, when the system

requirements are not very clear. Therefore, system architecture decisions need to be verified not just with drawings but also through code. Proving the architecture with code was not a common practice in the projects we assessed.

Automated tests play a significant role in terms of increasing the confidence of development teams with every change made progressing the development towards achieving higher quality software. For the three projects (A, B and C), the software and the system were manually tested. Automated unit and regression tests which are essential for continuous integration, have already started to be implemented in Company D as the goal is to achieve continuous delivery in the long term.

(b) Value Stream Mapping

In Figure 2, we provide an example of a value stream mapping study that we conducted within Company A. In this study, we looked at a feature's life cycle at a deep level and identified LT, PT and C/A%. The VSM study showed that the process time for a feature to complete was 2 weeks and 5 days whereas the lead time is 12.5 weeks. The result of the study made a significant impact on the group showing that there is a 10-week gap between Process Time and Lead Time. Upon analysis, the reason for the gap was attributed mostly to waiting rather than rework. After the study, requirements elicitation and analysis were selected as the first two areas to work.

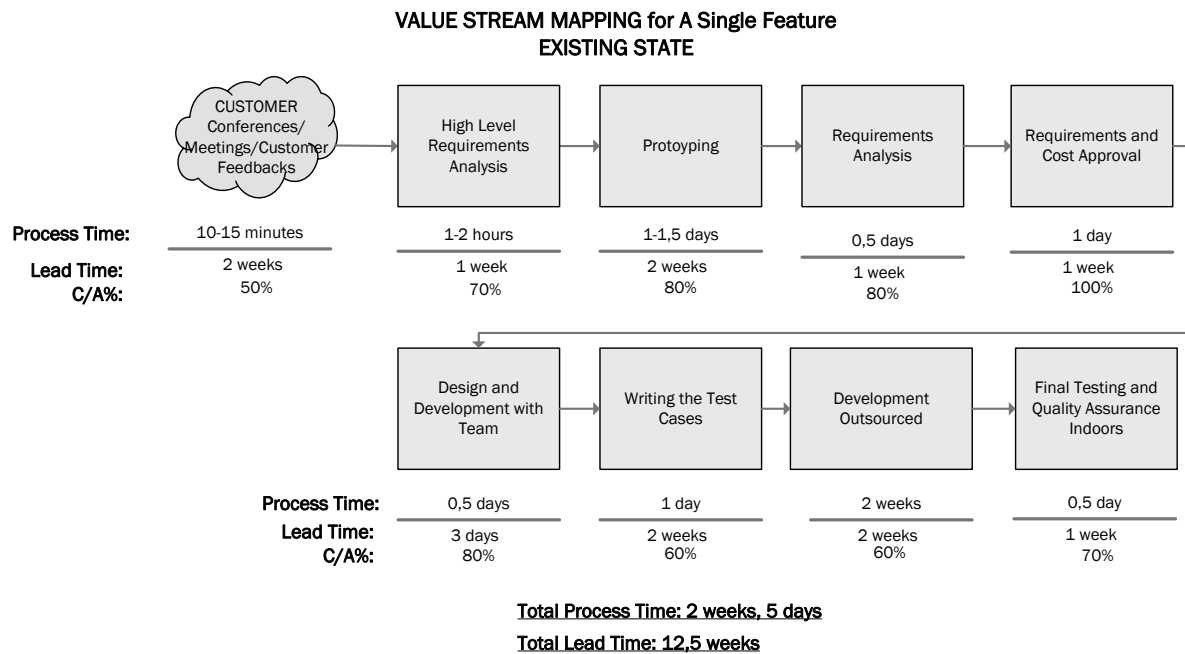


Figure 2 Value Stream Mapping for the Delivery of a Feature in Company A

(c) Prioritization

For each company, we provided a list of the issues found during the assessment and presented these issues to relevant parties within the Companies. Given the issues and the differences between process times and lead times in value stream mapping, we were able to focus our attention to critical processes that needs to be treated first.

Three highly connected processes in Company A, B and C which are stakeholder requirements definition, system requirement and software requirements, were selected as the first three processes to be focused. In parallel, it is aimed to establish automated unit test suites in parallel to other improvement studies.

In Company D, we decided to direct our focus on improving the communication among feature/product teams within the program to achieve better results in the areas of requirements management and software architecture, given there might be common or conflicting requirements that will affect the software architecture at the program level.

(d) **Improvement KATA:** Here, we provide an example of utilizing the Improvement KATA technique in Company C.

This section has been taken out of the pre-print version due to copy right issues. The full paper could be found at <https://onlinelibrary.wiley.com/doi/full/10.1002/smr.1929>.

(e) **Agile Recommendations**

This section has been taken out of the pre-print version due to copy right issues. The full paper could be found at <https://onlinelibrary.wiley.com/doi/full/10.1002/smr.1929>.

8. CONCLUSION and FUTURE PLANS

Medical device development organizations are seeking ways to be more flexible and adaptive, and ensure that they build safe and high-quality software and comply with the regulatory requirements. Based on both the safety classification and region they wish to market their medical device such organizations need to comply with different types of regulatory standards. In order to solve the complexity problem regarding compliance to different regulatory standards, we previously developed MDevSPICE[®]. Within this paper, we highlighted that whilst MDevSPICE[®] works well for companies developing software in a plan driven environment, the recommendations were difficult to implement within an agile development approach. Therefore, to solve this problem, we proposed a hybrid assessment approach. The approach includes the integration of agile practice recommendations with MDevSPICE[®] process assessments, then practicing value stream mappings as part of the prioritization phase, and then implementing suggested changes via the Improvement KATA technique.

In this paper, we described the hybrid assessment approach for medical device companies and its implementation in four medical device software development companies. We performed onsite process assessments based on MDevSPICE[®], analyzed their value streams, provided recommendations and prioritized improvement actions in terms of processes using action

research. For these four companies, we identified and discussed which processes contained gaps and suggested improvement practices.

Discussions with the assessed companies during the findings presentations reflected that based on the gaps and recommendations we presented to those companies, that the companies were keen to adopt this approach. Therefore, we feel that there is a need to build upon this research. In future studies with the companies, we will continue implementing agile specific improvement actions in an experimental way and periodically evaluating the progress. Value stream mapping studies will be repeated in three-month intervals to observe improvement in the lead times of the software development life cycle activities.

In terms of the assessment process, we observed that it is more valuable within a short timeframe, to get a higher-level view of the project assessed across more processes, rather than completing a more detailed assessment across fewer processes. As the initial goal of the assessments performed was not for the purpose of preparing companies for regulatory audits but rather for the identification of gaps, and to recommend improvement needs, and for the prioritizations of the future actions.

In terms of refining the hybrid assessment approach, an agile practices repository will be composed to provide two-way traceability between agile practices and MDevSPICE® base practices. We will be repeating this action research with other medical device software development companies and reflecting upon the lessons learned from the field studies relating to the hybrid assessment approach.

Acknowledgment

This research is supported by the Science Foundation Ireland Research Centres Programme, through Lero - the Irish Software Research Centre (<http://www.lero.ie>) grant 10/CE/I1855 & 13/RC/2094.

9. References

- [1] M. Lepmets, F. McCaffery, and P. Clarke, "Development and benefits of MDevSPICE®, the medical device software process assessment framework," *Journal of Software: Evolution and Process*, vol. 28, no. 9, pp. 800-816, 2016.
- [2] M. Rother, *Toyota Kata: managing people for improvement, adaptiveness and superior results*. McGraw-Hill Professional, 2009.
- [3] *FDA Chapter I - Food and drug administration, department of health and human services subchapter H - Medical devices, Part 820 - Quality system regulation*, 2016.
- [4] *Council Directive 1990. Council directive 90/385/EEC on active implantable medical devices (AIMDD)*.
- [5] *European Commission 1993. Directive 93/42/EEC of the European Parliament and of the Council concerning medical devices*.
- [6] *European Commission 1998. Directive 98/79/EC of the european parliament and of the council of 27 october 1998 on in vitro diagnostic medical devices. Official Journal of the European Communities .*
- [7] *EC 2007. Directive 2007/47/EC of the European Parliament and of the Council concerning medical devices. Official Journal of the European Union. Official Journal of the European Union.*
- [8] *IEC 2006. IEC 62304: Medical Device Software - Software Life-Cycle Processes.*
- [9] *ISO 2009. ISO 14971 - Medical Devices - Application of Risk Management to Medical Devices.*
- [10] *IEC 2009. IEC TR 80002-1 - Medical Device Software - Part 1: Guidance on the Application of ISO 14971 to Medical Device Software.*
- [11] *ISO 2003. ISO 13485: Medical Devices - Quality Management Systems - Requirements for Regulatory Purposes.*
- [12] *IEC 60601-1 - Medical electrical equipment – Part 1: General requirements for basic safety and essential performance 2005.*
- [13] *IEC 62366 - Medical devices - Application of usability engineering to medical devices, 2007.*
- [14] *IEC 2012. IEC 82304-1: Health Software – Part 1: General requirements for product safety.*
- [15] *FDA 2015. Guidance Document - Medical Devices and Radiation-Emitting Products. Center for Devices and Radiological Health.*
- [16] "FDA 2005. Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices," ed.
- [17] *FDA 2002. General Principles of Software Validation ; Final Guidance for Industry and FDA Staff.*
- [18] *FDA 2013. Mobile Medical Applications Guidance for Industry and Food and Drug Administration Staf.*
- [19] M. Lepmets, F. Mc Caffery, and P. Clarke, "Piloting MDevSPICE: the medical device software process assessment framework," in *Proceedings of the 2015 International Conference on Software and System Process*, 2015, pp. 9-16: ACM.
- [20] *ISO/IEC 12207:2008 Systems and software engineering -- Software life cycle processes, 2008.*

- [21] IEC 2014. *IEC TR 80002-3: Medical device software - Part 3: Process reference model of medical device software life cycle processes (IEC 62304)*.
- [22] *Capability Maturity Model Integrated-Development*, 2010.
- [23] *ISO/IEC 15504-5:2012 Information technology -- Process assessment -- Part 5: An exemplar software life cycle process assessment model*, 2012.
- [24] D. J. Anderson, "Stretching Agile to fit CMMI level 3," *Microsoft Corporation*, 2005.
- [25] M. Fritzsche and P. Keil, "Agile Methods and CMMI: Compatibility or Conflict?," *e-Informatica Software Engineering Journal*, vol. 1, no. 1, pp. 9-26, 2007.
- [26] J. A. H. Alegria and M. C. Bastarrica, "Implementing CMMI using a combination of Agile Methods," *CLEI Electronic Journal*, vol. 9, no. 1, 2006.
- [27] H. Glazer, J. Dalton, D. Anderson, M. D. Konrad, and S. Shrum, "CMMI or Agile: Why Not Embrace Both!," 2008.
- [28] B. Boehm, "Get ready for agile methods, with care," *Computer*, vol. 35 no. 1, pp. 64-69, 2002.
- [29] R. Turner and A. Jain, "Agile meets CMMI: Culture clash or common cause?," *Extreme Programming and Agile Methods—XP/Agile Universe 2002*, pp. 153-165, 2002.
- [30] (2001). *Agile Manifesto*. Available: www.agilemanifesto.org
- [31] S. W. Ambler and M. Lines, *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*. IBM Press, 2012.
- [32] J. Sutherland and K. Schwaber, "The scrum guide," *The Definitive Guide to Scrum: The Rules of the Game. Scrum.org*, 2013.
- [33] K. S. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley Professional, 2012.
- [34] A. Cockburn, *Agile software development: the cooperative game (agile software development series)*. Addison-Wesley Professional, 2006.
- [35] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods: Review and analysis," ed: VTT Finland, 2002.
- [36] B. Fitzgerald, K.-J. Stol, R. O'Sullivan, and D. O'Brien, "Scaling agile methods to regulated environments: An industry case study," in *Software Engineering (ICSE), 2013 35th International Conference on*, 2013, pp. 863-872: IEEE.
- [37] S. Wolff, "Scrum goes formal: Agile methods for safety-critical systems," in *Proceedings of the First International Workshop on Formal Methods in Software Engineering: Rigorous and Agile Approaches*, 2012, pp. 23-29: IEEE Press.
- [38] S. W. Ambler. (2002). *When Does(n't) Agile Modeling Make Sense?* Available: <http://www.agilemodeling.com/essays/whenDoesAMWork.htm>
- [39] R. Rasmussen, T. Hughes, J. Jenks, and J. Skach, "Adopting agile in an FDA regulated environment," in *Agile Conference, 2009. AGILE'09.*, 2009, pp. 151-155: IEEE.
- [40] K. Manjunath, J. Jagadeesh, and M. Yogeesh, "Achieving quality product in a long term software product development in healthcare application using Lean and Agile principles: Software engineering and software development," in *Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on*, 2013, pp. 26-34: IEEE.
- [41] O. Cawley, I. Richardson, X. Wang, and M. Kuhrmann, "A conceptual framework for lean regulated software development," in *Proceedings of the 2015 International Conference on Software and System Process*, 2015, pp. 167-168: ACM.
- [42] M. Mc Hugh, O. Cawley, F. McCaffcry, I. Richardson, and X. Wang, "An agile v-model for medical device software development to overcome the challenges with plan-driven software development lifecycles," in *Software Engineering in Health Care (SEHC), 2013 5th International Workshop on*, 2013, pp. 12-19: IEEE.
- [43] K. Łukasiewicz and J. Górski, "AgileSafe-a method of introducing agile practices into safety-critical software development processes," in *Computer Science and Information Systems (FedCSIS), 2016 Federated Conference on*, 2016, pp. 1549-1552: IEEE.

- [44] P. Anitha, D. Savio, and V. Mani, "Managing requirements volatility while "Scrumming" within the V-Model," in *Empirical Requirements Engineering (EmpiRE), 2013 IEEE Third International Workshop on*, 2013, pp. 17-23: IEEE.
- [45] P. Morrison, C. Holmgren, A. Massey, and L. Williams, "Proposing Regulatory-Driven Automated Test Suites," in *Agile Conference (AGILE), 2013*, 2013, pp. 11-21: IEEE.
- [46] K. Könnölä, S. Suomi, T. Mäkilä, T. Jokela, V. Rantala, and T. Lehtonen, "Agile methods in embedded system development: Multiple-case study of three industrial cases," *Journal of Systems and Software*, vol. 118, pp. 134-150, 2016.
- [47] R. Oshana, *Software engineering for embedded systems: Methods, practical techniques, and applications*. Newnes, 2013.
- [48] A. f. t. A. o. M. I. (AAMI), "AAMI TIR45:2012 Guidance on the use of AGILE practices in the development of medical device software."
- [49] J. Humble, J. Molesky, and B. O'Reilly, *Lean enterprise: How high performance organizations innovate at scale*. " O'Reilly Media, Inc.", 2014.
- [50] K. Beck, *Extreme programming explained: embrace change*. Addison-Wesley Professional, 2000.
- [51] K. Schwaber, "Scrum development process," in *Business Object Design and Implementation*: Springer, 1997, pp. 117-134.
- [52] P. Swartout, *Continuous Delivery and DevOps—A Quickstart Guide*. Packt Publishing Ltd, 2014.
- [53] M. Poppendieck and T. Poppendieck, *Lean software development: An agile toolkit*. Addison-Wesley Professional, 2003.
- [54] E. Brechner, *Agile Project Management with Kanban*. Pearson Education, 2015.
- [55] D. Leffingwell, *SAFe® 4.0 Reference Guide: Scaled Agile Framework® for Lean Software and Systems Engineering*. Addison-Wesley Professional, 2016.
- [56] K. Martin and M. Osterling, *Value stream mapping: how to visualize work and align leadership for organizational transformation*. McGraw-Hill Education, 2014.
- [57] M. Rother and J. Shook, *Learning to see: value stream mapping to add value and eliminate muda*. Lean Enterprise Institute, 2003.
- [58] D. E. Avison, F. Lau, M. D. Myers, and P. A. Nielsen, "Action research," *Communications of the ACM*, vol. 42, no. 1, pp. 94-97, 1999.
- [59] K. J. McDonald, *Beyond requirements: analysis with an agile mindset*. Addison-Wesley Professional, 2015.
- [60] J. Patton and P. Economy, *User story mapping: discover the whole story, build the right product*. " O'Reilly Media, Inc.", 2014.

