

# An Agile V-Model for Medical Device Software Development to Overcome the Challenges with Plan-Driven Software Development Lifecycles

Martin McHugh<sup>1</sup>, Oisín Cawley<sup>2</sup>, Fergal McCaffery<sup>1</sup>, Ita Richardson<sup>2</sup>, and Xiaofeng Wang<sup>3</sup>.

<sup>1</sup>RSRC, Dept. of Computing & Mathematics,  
Dundalk Institute of Technology & Lero,  
Co. Louth, Ireland  
{martin.mchugh, fergal.mccaffery}@dkit.ie

<sup>2</sup>Lero – The Irish Software Engineering  
Research Centre, University of Limerick,  
Co. Limerick  
Ireland  
{oisin.cawley, ita.richardson}@lero.ie

<sup>3</sup>Free University of Bozen,  
Bolzano, Italy  
xiaofeng.wang@unibz.it

**Abstract**—Through the use of semi structured interviews with medical device software organizations it emerged that medical device software organizations are experiencing difficulties when following plan driven Software Development Life Cycles (SDLC), particularly in the area of requirements management. To attempt to resolve these issues an examination of the non-regulated industry was performed to determine if lessons learned there could be applied to the development of medical device software. This examination revealed that agile methods are being widely adopted in the non-regulated software industry. To learn if agile methods could be adopted when developing medical device software a mapping study was performed which looked for instances of where agile methods have been used in regulated industries and where they have been adopted, to what success. This mapping study revealed that incorporating agile practices with the existing plan driven SDLC is the most favourable choice for medical device software organizations. This research aims to develop a SDLC which has a foundation of a plan driven SDLC which incorporates agile practices which can be followed when developing regulatory compliant software.

**Index Terms**—Medical Device Software, Safety Critical, V-Model, Agile, FDA, AV-Model

## I. INTRODUCTION

Medical device software, as with most other safety critical software, must be developed in accordance with the regulatory requirements of the region into which the software is being marketed. Regulatory requirements are in place to ensure the safe and reliable functioning of the software. For the purpose of this paper, we will be focusing on developing medical device software which must be compliant with the Federal Drugs Administration (FDA) regulations for use in the United States (US).

Medical device software developers typically develop software in accordance with a plan driven sequential Software Development Lifecycle (SDLC), such as the V-Model [1], as it appears to be the “best fit” with regulatory requirements [2]. To gain a deeper insight into the challenges experienced when developing medical device software we performed semi-structured interviews with 7 medical device software organizations. An expected finding of the interviews was that regulatory controls introduce a large amount of overhead when

developing medical device software. Each of the organizations identified that the process of introducing a change once development has begun results in revisiting stages of development and therefore increases costs. Based upon the findings of the interviews we began to look at software development methods that are better suited to accommodating requirements changes

In the non-regulated software development industry, there is a move away from plan driven SDLC’s towards agile software development methodologies and practices [3]. Agile methodologies and practices appear to offer a “silver bullet” to problems associated with following a plan driven SDLC such as being inflexible and unable to accommodate changes [4, p.41]. Evidence exists which reports significant benefits being gained within the non-regulated software development industry from utilising agile practices, such as reduced costs, reduced time to market and increased quality [5]. Despite this, the rate of adoption of agile practices amongst medical device software development organizations is low. There is no conclusive reason for this; however, a number of reasons have been cited. One such reason is that agile practices appear to be contradictory to regulatory requirements [6]. For example, medical device software organizations are required to submit extensive documentation to prove their device is safe for use in order to achieve regulatory approval, yet one of the four values of the agile manifesto states, “working software over comprehensive documentation” [7]. This would appear to suggest that following agile methods would not produce the necessary documentation required when seeking regulatory approval. Even though agile software development practices are often perceived to be contradictory to regulatory requirements, case studies have emerged from medical device software development organizations which have successfully adopted agile practices and received regulatory approval. For example, Abbott Diagnostics integrated agile practices with their plan driven SDLC on a development project and reported a cost saving of between 35% and 50% when compared to following a plan driven software development lifecycle [8].

Lightweight methods such as the Crystal Family of methodologies do appear to offer guidance for the development of safety critical software. Crystal methods place the agile values secondary to the primary focus of the software

which can be: *People, Interactions, Community, Skills, Talents and Communications*. The Crystal methods use a coloured weighting scheme based upon criticality and objectives to determine which “colour” to use i.e. Crystal Clear, Crystal Orange etc. [9]. This would appear to be of value to the development of medical device software; however, the crystal methods do not take into account regulatory requirements and therefore can be difficult to wholly follow when developing regulatory compliant software.

This paper attempts to answer the following research questions:

- 1) *Can agile practices be used to develop medical device software?*
- 2) *If agile practices can be used to develop medical device software, how must be incorporated with the existing lifecycle in order to meet regulatory requirements?*

To answer these research questions, semi structured interviews and a mapping study were performed. This paper outlines the interviews conducted with medical device software organizations as part of this research into the field of medical device software development. In our discussion on the SDLC which medical device software organizations currently adopt (Section II-C), we note that prior research has found that the V-Model is the most widely used [1]. Following this, we carried out a Mapping Study into the use of agile software development in medical device software development industries and propose the Agile V-Model (AV-Model) as a SDLC for use in the medical device industry. It aims to provide medical device software developers with the structure to follow a plan driven approach whilst reaping the benefits of utilising agile practices.

## II. MEDICAL DEVICE SOFTWARE DEVELOPMENT

### A. Software Development Techniques

Medical devices marketed for use within the USA must conform to the FDA CFR 21 Part 820 QSR [10]. Part 820.30 Subpart C “Design Controls” is specifically targeted at medical device software developers. The design controls cover the following areas: Design and Development Planning, Design Input, Design Output, Design Review, Design Verification, Design Transfer, Design Changes and Design History File.

Whilst these stages appear to follow each other sequentially, the FDA does not dictate the use of a sequential SDLC such as the Waterfall Model to complete these stages. The GPSV states “*this guidance does not recommend any specific life cycle model or any specific technique or method*”.

### B. Use of the V-Model in the Medical Device Industry

Despite not dictating a SDLC to follow, medical device software development organizations typically follow the V-Model [11]. It was first presented in 1991 at the NCOSE symposium [12] and is a variation on a SDLC which Royce presented which later became known as the Waterfall Model [13]. The V-Model identifies that there are different types of testing such as modular testing and integration testing [14]. The V-Model shows the relationship between the two sides of

the development process. This relationship is used to determine whether the stage has been completed successfully. If a problem occurs during the verification or validation of any one stage, then the opposite stage on the “V” must be revisited and if necessary reiterated [15]. Essentially, the testing of a product is planned in parallel with the corresponding phase of development. This method of developing software eases the process of achieving traceability. The FDA mandates that traceability be an integral part of a development process [16]. Therefore the V-Model is perceived to be the “best fit” with the regulatory requirements. While it may be the best fit, in practice the V-Model presents the same problems that are associated with utilizing any sequential plan driven SDLC. Royce, who presented the Waterfall model stated there are inherent problems associated with following a sequential lifecycle [13]. For example, as requirements are fixed at such an early stage, it can be very difficult to introduce a change in requirements once the project is underway. Also, it can be very difficult to capture all of the requirements at such an early stage of a project [4]. In addition to this, any changes introduced once a project is underway can create cost and budget overruns [17].

### C. AAMI TIR 45:2012

In October 2012, the Association for the Advancement of Medical Instrumentation (AAMI) released a Technical Information Report (TIR) entitled TIR 45:2012--Guidance on the use of agile practices in the development of medical device software [18]. The committee which developed the TIR consisted of industry experts and FDA staff. The AAMI recognised the shift in the non-regulated software development industry towards more agile practices and the evidence presented from successful adoption of agile practices in medical device software development organizations. However, they identified that the available information which details the use of agile practices in the development of medical device software was hard to understand and the objective of the TIR is to provide clear guidance on which agile practices are suited to the development of medical device software. The TIR also provides recommendations for complying with international standards and FDA guidance documents when using agile practices to develop medical device software. However, this document is at a high level and only addresses the use of a limited number of agile practices when developing software in accordance with IEC 62304 which in itself does not provide guidance for the development of standalone software [19].

## III. AGILE IN MEDICAL DEVICE SOFTWARE DEVELOPMENT

Research to date has revealed that the non-regulated software development industry is benefiting from moving towards more agile processes [8]. However, the medical device software development industry has not reaped the same benefits. This research aims to establish why these successes have not been replicated to great extent. This research is being performed following a Pragmatic Approach and in collaboration with medical device software organizations. As this research aims to overcome the challenges faced by these organisations it is therefore prudent to learn first-hand what the challenges

experienced are. This interviews detailed do not aim to seek statistical generalisation, but rather to seek theoretical generalisation [20].

### A. Semi-Structured Interviews

To gain a deeper insight into the challenges faced by medical device software development organizations Semi-Structured interviews were performed in accordance with Wengraf [21]. The interviews performed are known as Semi-Structured Depth Interviews (SSDI). SSDI are broken into two classifications, Heavily Structured Depth Interviews and Lightly Structured Depth Interviews. The degree of the structuring is determined by the degree to which the questions and interventions are pre-prepared by the researcher. In accordance with Wengraf the interviews were broken in to four elements: Research Purposes (RP), Central Research Questions (CRP), Theory Questions (TQ) and Interview Intervention (II)\Interview Questions (IQ).

The RP is the motivation behind the research being conducted. For this research, the RP is to gain a deeper insight into difficulties experienced when developing medical device software. The CRQ is the primary question(s) to which answers are being sought as a result of the interview being conducted. The TQ are high level questions. These questions are not asked directly to the interview participant. TQ are used to formulate the actual questions that will be asked of the participant. II/IQ is what is actually asked of the participant during the interview. The information gleaned from the responses is compiled to answer the TQ which in turn answer the CRQ which ultimately supports the RP.

The results of the interviews were analyzed in accordance with Wengraf’s Interview Material to Answers to Theory Questions to an Answer to the Central Research Question (IM-ATQ-ACRQ) model [21]. Whilst the CRQ > TQ > IQ/II model utilizes a top down approach, the IM-ATQ-ACRQ model utilizes a bottom up approach to determine the answer to the central research question. This method was used as it complimented the method employed for the creation of the interview questions i.e. RP > CRQ > TQ > IQ/II. The results were also analyzed in accordance with Miles and Huberman’s [22] method of analyzing qualitative data i.e. Data Reduction, Data Display and Conclusion Drawing & Verification.

Each of the organizations involved in the interviews identified that a major problem they experience is accommodating changes once development has begun. To accommodate changes a number of stages may need to be revisited, having a knock on effect of increasing rework and therefore increasing cost. When asked in the interviews how to resolve the problems associated with changing requirements a number of responses were given. One organization suggested the establishment of an incubation period prior to the requirements analysis stage. This incubation period would allow the customer time to consider all potential features they wished to include in the software and ideally removing the need for a change to be implemented once the project has begun. Another organization suggested placing greater emphasis on up-front planning and again making sure all of the necessary requirements were captured. One organization

suggested “placing manners on the customer” and preventing them from introducing a change once development has begun.

Each of these suggestions has their own merit, however these are proactive steps, none of the organizations were able to suggest a reactive response to when a requirements change was unavoidable. Current plan driven SDLCs are rigid and therefore have difficulty accommodating a change. Typically, when a change is introduced a number of stages need to be revisited to accommodate the change. This can require a lot of rework therefore increasing cost and development time. As a result a software development method which can accommodate changes once development has begun could bring benefit to medical device software organizations.

### B. Mapping Study

While the interviews performed identified the challenges faced by medical device software development organisations, they did not answer the question as to how these problems can be overcome. As discussed agile methods would appear to solve the problems mentioned; however this needed to be confirmed. To understand why the medical device software development industry has not benefited from adopting agile practices this paper focuses on the following research questions.

To answer these questions, a Mapping Study was performed. This Mapping Study was conducted in accordance with guidelines from Petersen et al. [23]. From an overall perspective, this process involved three main steps: 1. planning the review, 2. conducting the review and 3. reporting the review.

By approaching the review in such a systematic manner and demonstrating the rigor applied to each step, we build a higher level of confidence in our conclusions. We developed a protocol containing a full breakdown of the research approach. The Mapping Study quickly showed there to be limited published material in this specific area. In order to progress our investigation, we widened our review to cover regulated safety-critical software development in general. Due to similarities in the domains, lessons learned from within other regulated safety critical industries, such as Avionics and Automotive, software development can potentially be applied to the development of medical device software. We specifically included the following priori assumption to make any bias clearly identifiable:

TABLE 1 LITERATURE SOURCES

SOURCE	URL
ACM	<a href="http://portal.acm.org/portal.cfm">http://portal.acm.org/portal.cfm</a>
Compendex	<a href="http://www.engineeringvillage2.org">http://www.engineeringvillage2.org</a>
IEEE	<a href="http://ieeexplore.ieee.org/Xplore">http://ieeexplore.ieee.org/Xplore</a>
INSPEC	<a href="http://www.theiet.org/inspec">http://www.theiet.org/inspec</a>
Science Direct	<a href="http://www.sciencedirect.com">http://www.sciencedirect.com</a>
Web of Science	<a href="http://apps.isiknowledge.com">http://apps.isiknowledge.com</a>
Misc.	Book Sections, Thesis, Industry Reports, Websites

TABLE 2 QUESTIONS USED TO SOLICIT QUALITY OF PAPER

Does the study predominantly relate to software development with a safety critical regulated setting?
Which particular agile methodology does it look at in depth?
Does it report on real-life case studies where an agile methodology was used?
Does the study find in favour of the applicability of agile software development methodologies within a safety-critical, regulated setting?
What 'flavour' of agile does the study promote? - Agile (the practices of a single agile methodology) - Agile – Agile (a combination of different agile methodologies) - Agile – Planned (A combination of agile and traditional methodologies) - None (it does not recommend agile methodologies)

*Some practices/aspects of agile software development do not adequately support all the requirements of the regulations for safety-critical software development, such as those laid down by the US regulatory bodies.*

The main tasks described in our protocol are: identifying data sources, building search strings, performing pilot search, adjusting search criteria, exporting results for citation management, applying inclusion/exclusion criteria, quality assessment, data extraction and data synthesis. While we found this a very positive approach in terms of providing a clear and unambiguous path through the literature, other important contributions were available from industry sources such as non-academic books, reports and other online resources. Our opinion is that this grey literature assists in addressing publication bias. Therefore, it was included in our study.

### C. Search Sources and Strings

Taking the Mapping Study guidelines as a starting point and looking at other published Mapping Studies and Systematic Literature Reviews (SLR) [24-27], we used the electronic sources noted in Table 1. An important lesson in the practice of searching electronic databases, is that each database search engine is different as highlighted by Brereton et al. [28].

The specific search strings were formed following the Population, Intervention, Comparison, Outcomes and Context (PICOC) criteria suggested by Petticrew and Roberts [29]. The generic query we used can be written as follows:

*“any document containing the phrase ‘medical device’ or ‘embedded software’ or the word stem ‘regulat’ AND any of the following phrases (‘software development’, ‘software process’, ‘software life-cycle’, ‘manufacturing software’) AND containing any of the following words/phrases (‘agile’, ‘scrum’, ‘XP’, ‘extreme programming’, ‘crystal’, ‘lean’).*

### D. Inclusion and Exclusion Criteria

The inclusion criteria were:

- Peer Reviewed Research Papers;

- Grey Literature from experienced practitioners;
- Relevant to safety-critical software development;
- Focus on agile practices.

The exclusion criteria included:

- Non-English language;
- Content too general (for example a high level review of agile practices);
- Duplicated work;
- Off Topic (Does not concentrate on expected subject matter)

### E. Quality Attributes

As described by Petersen et al. [23], an integral part of a Mapping Study is to assess the quality of the primary studies. This aids in determining the relevant importance of a study by capturing, for example:

- The possible effects of bias;
- The importance of a study in the body of literature;
- The relevance to your particular research questions(s).

In order to determine the value of each paper, a series of questions were asked of each one (Table 2) and the answer recorded on the data extraction sheet. Following this, we added an overall rating [1-poor to 5-excellent] to each entry. This step had particular importance as many of the papers were not empirically based. The average overall rating was 3.46. Of the 26 papers identified, 2 was the lowest rating given.

### F. Data Extraction

Following the technique used by Dyba and Dingsor [30] for citation management, the results of the searches were exported. The publications underwent an initial inclusion/exclusion analysis. This was carried out by the primary reviewer and validation performed by the secondary reviewers.

A two-step approach to the inclusion/exclusion was carried out. First, papers were excluded on the basis of their title and abstract. All remaining papers were then read in full and irrelevant papers were excluded. The remaining entries formed the basis of the review. The relevant data was extracted into Microsoft Excel Spread sheet (data extraction template), where subsequent information, such as inclusion/exclusion justification, quality attributes and a short note on the limitations of each publication, were recorded. From an initial count of 193 results, the data set was reduced to 64 in the first

TABLE 3 SEARCH RESULTS SUMMARY

SOURCE	Count	1 <sup>st</sup> Review	2 <sup>nd</sup> Review
ACM	72	18	6
Compendex	35	11	6
IEEE	20	11	7
INSPEC	15	5	2
Science Direct	29	4	0
Web of Science	5	0	0
Misc.	17	15	5

review, and then to 26 in the second review. The 26 results spanned the years 2002-2012. Of experience reports published at international conferences (50%), only 6 were classified as empirical research papers – 5 case studies and 2 surveys. This is indicative of the lack of empirical research in this specific field.

### G. Data Synthesis

While the Mapping Study detailed here looked at regulated embedded software development in general, we have a special interest in medical device software development. The Mapping Study found that 11 papers (42%) report from a medical device perspective. This acts as evidence that agile practices can be followed when developing medical device software. All of the organizations which reported using agile practices to develop medical device software, highlighted that using them had a positive impact within their development project. The agile methodologies which appeared most throughout the literature were XP and Scrum.

One of the areas we were interested in investigating was the ‘flavour’ of agile being adopted/trialled in this domain. The Mapping Study determined whether full standalone agile methodologies (Agile), a combination of different agile methodologies (Agile-Agile), or utilising agile in conjunction with traditional plan driven development techniques (Agile-Planned) was being favoured. The results show that almost 46% of the papers reported on the adoption or trials of Agile-Planned usage, with 19% adopting Agile, a further 19% adopting Agile-Agile and the remaining 16% reported no preference. Therefore, our Mapping Study provides evidence that a hybrid model incorporating agile with plan driven methodologies is the most favourable option when developing medical device software.

## IV. TAILORED SOFTWARE DEVELOPMENT LIFECYCLE

As shown in the Mapping Study, there is still a limited amount of publicly available information detailing where medical device organizations have utilised agile practices. Despite this, there is evidence to suggest [8, 31] that medical device software development projects can benefit from embracing agile practices. Whilst the Mapping Study revealed that agile practices can be used in the development of medical device software, it also revealed that a combination of Agile and Planned appears to be more suited to the development of medical device software. Through the Mapping Study, it emerged that there is currently no single SDLC which medical device software developers can follow when developing medical device software which combines agile and plan driven techniques. Instead, organizations employing agile are tailoring their existing lifecycles to incorporate agile practices. This method may be suited to large organizations with multiple projects that can trial agile practices in a project to determine whether they appropriate. Smaller medical device software developers may only be working on a single project at a time and cannot risk trialling agile on their only project. The hybrid SDLC proposed in this paper aims to provide medical device software development organizations, regardless of size or maturity, guidance on the development of

safe and reliable medical device software which is regulatory compliant, whilst reaping the benefits associated with utilising agile practices.

### A. Development of Agile V-Model

The process of developing the Agile V-Model is broken into clear distinct phases:

1. Selection of foundation plan driven SDLC;
2. Preparing for inclusion of agile practices into plan driven SDLC;
3. Identification of applicable agile practices to the development of medical device software.

1) *Selection of foundation plan driven SDLC:* When selecting the foundation of the hybrid SDLC, a number of plan driven SDLCs were examined. The conclusion was made that the V-Model is the most appropriate model on which to build the hybrid SDLC. The reasons for choosing the V-Model are:

- Medical device software organizations typically follow the V-Model to develop medical device software. As a result, they are already familiar with the structure and phases of the V-Model and would be more willing to adopt a hybrid model based upon a SDLC with which they are familiar.
- Medical device software organizations may have received regulatory approval to follow the V-Model when developing medical device software. If these organizations move to a completely different SDLC, they may need to re-apply for regulatory approval for the new SDLC. This may be a barrier as organizations could be reluctant to undergo regulatory approval again.
- Whilst none of the regulatory requirements or development standards mandate the use of the V-Model, it appears to be the best fit with regulatory requirements, as it guides organizations through the process of producing the necessary deliverables required to achieve regulatory conformance.

2) *Preparing for Inclusion of Agile practices into plan driven SDLC:* Each of the agile methodologies advocates iterative software development. Each of the sequential plan driven SDLCs suffer the problem of being rigid and inflexible to change. With iterative techniques, changes can be introduced to a development project without needing to revisit a number of other stages of the SDLC. However, to incorporate iterative techniques, the process of “Risk Identification” needs to be added to the model. Risk Identification involves analysing the project, dividing it into iterations and identifying the iterations which pose the most risk to the project. The iterations that pose the most risk are then performed as early as possible in the project. Once risk identification is added, each of the stages of the V-Model is assessed to determine which stages could be performed iteratively. As a result, all of the stages of the development lifecycle are divided into two categories: stages that can be performed iteratively and stages that can only be performed in a single pass. For example, the FDA requires the device

manufacturers to submit high level requirements prior to beginning development. Therefore, this can only be done once. Also, the process of achieving regulatory approval can only be sought when a device is completed and the acceptance tests have all passed. Therefore, this can only be completed once. However, other stages such including “Software Architecture Design” and “Unit Implementation” can be performed iteratively.

3) *Identification of applicable agile practices to the development of medical device software;* To identify agile practices applicable to the development of medical device software, each of the agile methodologies - Scrum, XP, DSDM and Crystal clear - were examined. Based upon this examination, 59 agile practices were identified. A comparison between these 59 practices and the appropriate regulations and standards was performed. This comparison revealed that none of the 59 identified practices contradict regulations or development standards. However, despite these practices not being contradictory to regulations or standards, their applicability to the development of medical device software development remains unclear. To determine the level of applicability, based upon the findings of the Mapping Study, 13 practices were identified as being applicable to the development of medical device software. These practices have been selected based on the fact that they have been successfully adopted in medical device software organisations developing regulatory compliant software [32]. These practices include iterative development, use cases/user stories and test driven development. These 13 practices were then mapped to the appropriate stage of the SDLC. A problem associated with following a plan driven SDLC, is the emphasis placed on up-front planning. This can result in a project suffering if a change is introduced after development has begun. Using iterative development, detailed requirements can

be easily revisited and if a change in requirements is made, this change can be accommodated in an upcoming iteration.

Whilst only 13 practices have been identified to date, we are continuing our validation on the remaining 46 practices. On-going research will determine how many of the remaining practices are applicable to medical device software development and these practices will be mapped to the model were appropriate. Some of the remaining practices to be examined for applicability include, Continuous Improvement, Definition of Done and Test Driven Development.

### B. Hybrid Model

Figure 1 shows the AV-Model. Each of the 13 practices identified through the Mapping Study is mapped to the appropriate stage of development such as “On Site Customer”, “Iterative Development”, “Use Cases / User Stories”. Whilst practices have been mapped to specific stages, it does not preclude the use of the practice at another stage of development. For example, Fig. 2 shows the use of “User

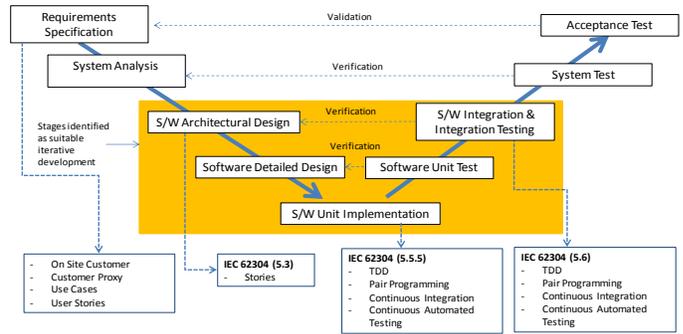


FIGURE 1 THE AV-MODEL

Stories” and “Use Cases” at the “Requirements Specification” stage. These can also be used in the “Architectural Design” stage. Two identified applicable practices not shown in the model are “Collocated Teams” and “Self Organising Teams”. These are not displayed graphically on the model as these practices do not relate to a specific stage of the development lifecycle. Rather, they are related to team structure.

### C. Model Validation

The objective of the development of this model is to resolve problems associated with following a plan driven software development lifecycle, whilst reaping the benefits of utilising agile practices. As the model is currently under development it has not yet been fully validated. However, there will be two stages in the process of validation: *Expert Opinion and Implementation*. The development of the AV model is an iterative one. Once a stage of validation is complete feedback will be applied and the model will proceed to the next stage of validation. Feedback will be obtained through the use of a survey instrument with open ended questions.

1) *Expert Opinion:* Once the model has received validation from industry it will be distributed to experts in the field of medical device software development. Agreement has already been made with members of the IEC 62304 committee and members of the TIR 45:2012 committee to provide validation of the model. By eliciting this form of feedback the model aims to gain acceptance in the standards community.

2) *Implementation:* Once the model has undergone the Expert Opinion validation, it will be adopted by a medical device software development organization. A medical device software organization has agreed to implement the model once it is completed and passed through each of the steps of validation.

## V. RELATED WORK

In [33] research is conducted that provides information as to how the avionics industry can benefit from adopting agile practices. In this research, the author investigates the regulatory constraints placed upon avionic software developers adhering to DO-178B [34] and whether or not XP can be used in this domain. The research revealed that XP could not practically be used in the case study, but the author

surmises that avionics software development could benefit from adopting agile practices. In [33] the authors provide information as to how employing XP can be beneficial in areas such as requirements managements and change management. However, the authors also discuss that for XP to be a success in such a project there are prerequisites such as early customer involvement.

In Vanderleest and Butler [35] a mapping is performed between Agile practices and the development practices as part of DO-178B. This mapping demonstrates the ability to utilize agile practices in the development of avionics software. However, Vanderleest identifies that there has not been a large amount of research in the area of using Agile in avionics and calls for other researchers to establish collaboration to research the area further.

Manhart and Schneider [36] present a case study of the adoption of agile practices in the Daimler Chrysler software engineering department. Within this study the organization examined the possibility of adopting a full Agile methodology, but found that a tailored framework suited there development requirements. To that end, elements of agile i.e. test first process, were integrated into a traditional process improvement model. This research did not focus on any one agile methodology to extract practices from, but rather took a wider view of all of the agile practices.

This related work shows that other regulated industries are examining the possibility of utilizing agile practices to overcome the challenges associated with following plan driven SDLCs. It also shows that where agile practices have been used they have been most successful when incorporated with a plan driven approach.

## VI. CONCLUSIONS

Through semi structured interviews it was revealed that medical device software organizations are experiencing difficulties when following a plan driven SDLC in areas such as requirements management. To overcome these problems an examination of other development techniques was performed. This examination revealed that agile methods could potentially overcome the challenges associated with following a plan driven SDLC. To that end, through the conduction of a Mapping Study, it has been shown that medical device software development organizations can develop regulatory compliant software whilst utilising agile practices. The Mapping Study also revealed that, where adopting agile has proved successful, the existing lifecycle was tailored to accommodate agile practices rather than wholly embracing a complete agile methodology such as Scrum or XP.

Although agile practices have been successfully adopted in medical device software development organizations, this success has not been replicated to a great extent. One potential reason for this is the reluctance of medical device software developers to move away from tried and tested techniques such as the V-Model. An additional reason may also be that medical device software development organizations have already achieved regulatory approval to use their current SDLC and, if they moved to a completely different SDLC,

they may need to submit for approval once more. The hybrid model, being based upon the V-Model, will remove this need.

This paper proposes a hybrid SDLC known as the AV-Model, which combines both agile and plan driven development practices can follow when developing regulatory compliant software. This SDLC has been developed to resolve some of the problems, such as inflexibility, which medical device software development organizations are experiencing when following a plan driven SDLC. Whilst the development of this model is on-going, medical device software development organizations can benefit from the results of this research to date and it is expected that the model will grow to incorporate additional applicable practices. The remaining applicable practices will be identified through collaborations with medical device software development organizations.

## ACKNOWLEDGMENTS

This research is supported by the Science Foundation Ireland (SFI) Stokes Lectureship Programme, grant number 07/SK/I1299, the SFI Principal Investigator Programme, grant number 08/IN.1/I2030 (the funding of this project was awarded by Science Foundation Ireland under a co-funding initiative by the Irish Government and European Regional Development Fund), and supported in part by Lero - the Irish Software Engineering Research Centre (<http://www.lero.ie>) grant 10/CE/I1855.

## VII. REFERENCES

- [1] F. McCaffery, V. Casey, M. Sivakumar, G. Coleman, P. Donnelly, and J. Burton, "Medical Device Software Traceability," in *Software and Systems Traceability*, J. Cleland-Huang, O. Gotel, and A. Zisman, Eds., ed: Springer-Verlag, 2012.
- [2] F. McCaffery, D. McFall, P. Donnelly, F. G. Wilkie, and R. Sterritt, "A Software Process Improvement Lifecycle Framework for the Medical Device Industry," presented at the Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05), 2005.
- [3] S. Ambler. (2008). *Agile Adoption Rate Survey Results: February 2008* Available: <http://www.ambysoft.com/surveys/agileFebruary2008.html#Downloads> [Last accessed: 23/09/2012]
- [4] J. Cadle and D. Yeates, *Project Management for Information Systems*: Pearson Education, 2008.
- [5] VersionOne, "State of Agile Survey - The Stage of Agile Development," 2010.
- [6] D. Vogel, "Agile Methods: Most are not ready for prime time in medical device software design and development," *DesignFAX Online*, 2006.
- [7] R. C. Martin, *Agile Software Development - Principles, Patterns and Practices*: Prentice Hall, 2003.
- [8] R. Rasmussen, T. Hughes, J. R. Jenks, and J. Skach, "Adopting Agile in an FDA Regulated Environment," presented at the Agile Conference, 2009. AGILE '09. , Chicago, IL 2009.
- [9] A. Cockburn. (2008). *Crystal Methodologies*. Available: <https://www.google.ie/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CC4QFjAA&url=http%3A%2F%2Falistair.cockburn.us%2FCrystal%2Bmethodologies&ei=no40UaenFdCYhQeB4oDYBA&usg=AFQjCNExy8tnYy5ogOW7SXIrU5hXgpIXDQ>
- [10] FDA, "Title 21--Food and Drugs Chapter I --Food and Drug Administration Department of Health and Human Services subchapter h--Medical Devices part 820 Quality System Regulation," ed: *U.S. Department of Health and Human Services*, 2007.
- [11] F. McCaffery, D. McFall, P. Donnelly, and F. G. Wilkie, "Risk Management Process Improvement for the medical device industry,"

- presented at the Conference on Software Development (SWDC-REK-2005) Iceland, 2005.
- [12] K. Forsberg and H. Mooz, "The Relationship of Systems Engineering to the Project Cycle," presented at the First Annual Symposium of the national Council on Systems Engineering (NCOSE), Chattanooga, Tennessee, 1991.
- [13] W. Royce, "Managing the Development of Large Software Systems," presented at the Proceedings of IEEE WESCON, 1970.
- [14] P. E. Rook, "Controlling software projects," *IEEE Software Engineering Journal*, vol. 1, p. 7, 1986.
- [15] S. L. Pfleeger and J. M. Atlee, *Software Engineering: Theory and Practice*. New Jersey: Pearson Higher Education, 2001.
- [16] V. Casey and F. McCaffery, "Med-Trace: Traceability Assessment Method for Medical Device Software Development.," presented at the European Systems & Software Process Improvement and Innovation Conference, (EuroSPI). Roskilde, Denmark, 2011.
- [17] N. M. A. Munassar and A. Govardhan, "A Comparison Between Five Models Of Software Engineering," *IJCSI International Journal of Computer Science Issues.*, vol. 7, pp. 94-101, 2010.
- [18] AAMI, "AAMI TIR45:2012 -- Guidance on the use of agile practices in the development of medical device software," 2012.
- [19] M. McHugh, F. McCaffery, and V. Casey, "Standalone Software as an Active Medical Device " presented at the The 11th International SPICE Conference Process Improvement and Capability determination, Dublin, 2011.
- [20] A. Lee and R. Baskerville, "Generalising Generalisability in Information Systems Research," *Information Systems Research*, vol. 14, pp. 221-243, 2003.
- [21] T. Wengraf, *Qualitative Research Interviewing*. London: Sage Publications, 2001.
- [22] M. Miles and A. M. Huberman, *Qualitative Data Analysis: An Expanded Sourcebook*. London: Sage, 1994.
- [23] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic Mapping Studies in Software Engineering," presented at the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), University of Bari, Italy, 2008.
- [24] L. Chen, A.B. Muhammad, and C. Cawley, "A Status Report on the Evaluation of Variability Management Approaches," presented at the 13th International Conference on Evaluation and Assessment in Software Engineering-EASE, Durham University, Durham UK, 2009.
- [25] D. Smitte, C. Wohlin, T. Gorschek, and R. Feldt, "Empirical evidence in global software engineering: a systematic review," *Empirical Softw. Engg.*, vol. 15, pp. 91-118, 2010.
- [26] E. Hossain, M. A. Babar, and H.-y. Paik, "Using Scrum in Global Software Development: A Systematic Literature Review," presented at the Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering, 2009.
- [27] S. Lane and I. Richardson, "Process models for service-based applications: A systematic literature review," *Inf. Softw. Technol.*, vol. 53, pp. 424-439, 2011.
- [28] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *J. Syst. Softw.*, vol. 80, pp. 571-583, 2007.
- [29] M. Petticrew and H. Roberts, *Systematic Reviews in the Social Sciences: A Practical Guide*. Hoboken, NJ: Wiley-Blackwell, 2005.
- [30] T. Dyba and T. Dingsa, "Empirical studies of agile software development: A systematic review," *Inf. Softw. Technol.*, vol. 50, pp. 833-859, 2008.
- [31] P. A. Rottier and V. Rodrigues, "Agile Development in a Medical Device Company," presented at the Proceedings of the 11th AGILE Conference. AGILE '08., Girona, Spain, 2008.
- [32] M. McHugh, "Integrating Agile Practices with Plan-Driven Medical Device Software Development," presented at the The 13th International Conference on Agile Software Development: XP 2012 Doctoral Symposium, Malmo Sweden, 2012.
- [33] A. Wils, S. Baelen, T. Holvoet, and K. Vlamincx, "Agility in the avionics software world," presented at the 7th International Conference on Extreme Programming and Agile Processes in Software Engineering, XP, 2006.
- [34] *DO-178B: Software considerations in airborne systems and equipment certification.*, RTCA, 1992.
- [35] S. H. Vanderleest and A. Buter, "Escape the waterfall: Agile for aerospace," presented at the Digital Avionics Systems Conference, 2009. DASC '09. IEEE/AIAA 28th Orlando, FL, 2009.
- [36] P. Manhart and K. Schneider, "Breaking the Ice for Agile Development of Embedded Software: An Industry Experience Report," presented at the Proceedings of the 26th International Conference on Software Engineering, 2004.