Integrating Agile Practices with a Medical Device Software Development Lifecycle

Martin Mc Hugh[1], Fergal Mc Caffery[1], Valentine Casey[1], Minna Pikkarainen[2]

[1] Regulated Software Research Group, Department of Computing and Mathematics
Dundalk Institute of Technology, Co. Louth Ireland
{Martin.McHugh, Fergal.McCaffery, Val.Casey}@dkit.ie

[2] VTT, Technical Research Centre of Finland
Minna.Pikkarainen@vtt.fi

Abstract

The rate at which agile software development practices are being adopted is growing rapidly. Agile software development practices and methodologies appear to offer the silver bullet which can solve the problems associated with following plan driven software development lifecycles. Agile software development practices offer the possibility of achieving lower development costs, increased efficiency and improved software quality. However, there is currently a low rate of publicly available information that suggests there is widespread adoption of agile practices within the medical device software domain. This is largely due to the fact that software developed for medical devices includes challenges not faced when developing non safety critical software. As a result of these challenges, medical device software is typically developed using plan driven software development lifecycles. However, such lifecycles are quite rigid and cannot accommodate changes easily. Previous research has revealed that medical device software development projects can benefit from adopting agile practices whilst still maintaining the discipline associated with following plan driven development lifecycles. This paper outlines the challenges faced by developers when developing medical device software and how short-comings in both agile and plan driven approaches can be resolved by following a mixed method approach to medical device software development.

Keywords

Medical Device, Agile, Plan Driven, SDLC, XP, Scrum, V-Model, TDD, FDA, MDD, 2007/47/EC, Software Process Improvement

1 Introduction

Software developers are under increasing pressure to develop software faster that not only meets a customer's requirements, but exceeds them [1]. Initially, iterative software development appeared to offer a solution by aiding faster development of systems. Iterative techniques evolved and a subset known as agile appeared [2]. Larmin and Basili [3] identified Dynamic Systems Development Model (DSDM) as being the first agile methodology. Each of the agile methodologies consists of a number of agile practices. These practices are procedures defined as being highly effective and efficient [4] such as sprint planning, an open office space, daily meetings and from Scrum, product backlogs [5]. The use of agile methodologies such as XP and Scrum, in traditional software development projects are becoming increasingly popular [6].

In some case studies, agile practices have been used in the safety critical field. For instance, in Motorola, a select number of XP practices were used during the development of safety critical systems [7]. In these cases, the use of XP practices was reported to have had a 53% improvement in average quality compared to the plan-driven software development projects. A key challenge in these

XP projects was to define how the changes affected the overall end date of the projects. Additionally, Drobka et al.[7] discusses the issue of documentation production. In the case of Grenning [8], the source code was not sufficiently documented for the whole system, and this resulted in the need for a high-level architecture document which provided class diagrams, scenarios and a process view of the system for developers. Furthermore, acceptance tests were not adequate to verify the traceability from end product to customer requirements. Therefore, additional verification review meetings were used to cover this gap in the verification process [7]. Grenning [8] describes how XP practices such as the planning games, small releases, simple design, test first development; refactoring, pair programming, collective ownership, continuous integration and 40 hour week are used in a large company developing safety critical systems. As a result of this analysis, he suggests that some XP practices, such as simple designs integrated with test first development and refactoring work quite well in the safety critical area. In that case, the managers were reported to be happy with the results of the use of XP practices. This was mainly due to the XP team's ability to readily produce working software instead of a high amount of documentation [8]. One of the biggest challenges revealed from this case was that resistance was caused mainly due to the decreased amount documentation. This documentation was needed, for example, to define product requirements, sustain technical reviews, support maintenance and describe interfaces. Based on the experiences of the XP projects it was still understood that the documentation was needed for maintenance and review purposes [8].

In spite of these experiences of adopting agile practices in the safety critical domain, there is still little evidence of agile practices being widely embraced when developing medical device software [9]. The reason for this is still unclear, but one potential reason is that safety critical systems must also meet the appropriate regulatory requirements [10]. As part of the Food and Drug Administration (FDA) regulations most items require formal approval and this is cited as being a reason why many medical device software developers follow a sequential software development model such as the Waterfall or V-Model [11]. Of the limited information available, medical device software projects can benefit from incorporating agile practices, but no single agile methodology can be followed when developing medical device software [12].

It is generally perceived that international medical device regulatory requirements recommend that software developed either as a component of a medical device or as a medical device should be developed in accordance with a plan driven software development lifecycle such as the Waterfall Model or the V-Model. However, the regulatory requirements and development standards do not enforce the usage of any particular software development lifecycle. Whilst no specific software development lifecycle is mandated by the regulations or standards medical device software developers typically develop software in accordance with the V-Model [13]. Following the V-Model produces the necessary deliverables required in order to achieve regulatory approval. The objective of this research is to develop a Software Development Life Cycle (SDLC) that accrues the benefits of utilising agile practices whilst still producing the necessary deliverables provided by following plan driven software development lifecycles. This paper will serve as the foundation for this research. The remainder of this paper will be structured as follows.

In section 2, the latest regulatory requirements which medical device software organisations must adhere to are outlined. In section 3, information is provided as to how regulatory bodies do not require the usage of a specific software development lifecycle and that regulatory bodies are only concerned with the specific requirements such as traceability rather than being concerned with how these requirements are fulfilled. In section 4, a mixed method software development lifecycle is proposed that incorporates agile practices with a plan driven lifecycle and finally in section 5, the conclusions and plans for future work are presented.

2   International Regulations

Medical device software developers must adhere to the regulatory requirements of the region in which a medical device is being marketed for use. Medical devices being marketed for use within the United States (US) must conform to the FDA requirements and medical devices being marketed for use within the European Union (EU) must conform to the Medical Device Directive (MDD) and its latest amendment.

2.1   European Medical Device Directive

On March 21st 2010 the latest amendment to the MDD 2007/47/EC came into effect [14]. As stated, conformance to the latest amendment to the MDD is mandatory for a medical device to be marketed for use in the EU. Competent Authorities and notified bodies within each EU member state are responsible for certifying medical device conformance with the MDD. Once a device is certified within an EU member state the device can be marketed into all of the member states.

The latest amendment to the MDD introduced a number of changes; the most significant with regards to software development is that standalone software can now be classified as an active medical device. Prior to this amendment software was considered a component of a hardware medical device. With the release of this amendment software can now potentially be the only component of a medical device and therefore subject to full regulatory scrutiny. With this change a greater emphasis falls onto the international standards which are followed during the development of medical device software. IEC 62304:2006 [15] is a harmonised standard as part of the MDD [16]. Medical device software developers are recommended to follow IEC 62304 and its aligned standards to receive guidance. However, these aligned standards can be difficult to follow and this is particularly relevant when a medical device only consists of software.

2.2   FDA Regulations

Within the US the FDA is responsible for ensuring medical devices certified for use are safe and reliable. The FDA provides guidance documents which medical device software developers are recommended to follow in order to achieve regulatory conformance. These documents include:

- General Principles of Software Validation (GPSV) [17];
- Medical Device Data Systems Rule [18];
- Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices [19];
- Draft Guidance for Industry and Food and Drug Administration - Mobile Device Applications [20];
- Guidance for Industry, FDA Reviewers and Compliance on Off-The-Shelf Software use in Medical Devices [21].

All medical device manufacturers wishing to achieve FDA approval must conform to FDA 21 CFR Part 820 Quality System Regulations (QSR) [22]. FDA 21 CFR Part 820 requires medical device manufacturers to provide sufficient evidence that a Quality Management System (QMS) was employed when developing a medical device. To accompany, this all medical device manufacturers wishing to achieve FDA approval must follow FDA General Controls [23]. These controls include provisions that relate to device registration, branding and banned devices.

3   Agile Practices in the Development of Medical Device Software

When exploring possible frameworks for developing medical device software, we discovered that developers in this domain often select a plan driven lifecycle such as the V-Model [24], rather than considering following agile practices with iterative or incremental development lifecycle. This may be due to the fact that the agile manifesto places greater emphasis on people rather than processes and also considers documentation as being a secondary output from a software development project [25]. In practice, formal communication, such as source code, test cases, and essential amount of documentation is also used in the agile software development, but not in the same way or to the same extent as in the plan-driven software development process [26]. Additionally, some of the agile principles suggest that business people and developers must work together daily and project information should be shared through informal, face-to-face conversation rather than through documentation [27].

However, regulatory bodies in both the US and Europe require medical device manufacturers to provide documentation which provides information on areas such as risk and hazard management and traceability between the requirements stage and the development stages of a software development project [28, 29]. While this is the case, both the FDA guidelines and the MDD do not mandate the use of a specific life cycle model. The FDA states:

"(the FDA) does not recommend the use of any specific software life cycle model. Software developers should establish a software life cycle model that is appropriate for their product and organization"

To accompany this the FDA also states that the selected life cycle model "*should cover the software from its birth to its retirement",* and also that the lifecycle model should support validation and verification [17]. As discussed medical device software development organisations are recommended to follow IEC 62304 when developing medical device software. IEC 62034 states:

> "*The (IEC 62304) standard does not require a particular software development life cycle model*"

To accompany this IEC 62304 also states;

> "*Whichever life cycle is chosen it is necessary to maintain the logical dependencies between process outputs*"

IEC 62304 does not mandate the use of a specific lifecycle and is more concerned with the regulatory requirements such as traceability and not with how these requirements are fulfilled. Both the FDA GPSV guidance document and the IEC 62034 standard discuss the use of different life cycle models including the waterfall, incremental and evolutionary models. The FDA GPSV states:

> "*Most software development models will be iterative. This is likely to result in several versions of both the software requirement specification and the software design specification*"

Currently, the Association of Advancement of Medical Instrumentation (AAMI) (the publishers of IEC 62304) is in the process of completing a technical information report which will provide recommendations as to how medical device software development organisations can comply with international and FDA regulations by using agile practices [30]. This document will provide a complete mapping between the 12 agile principles [31] and each of the development stages of IEC 62304. All of this information signals that regulatory bodies acknowledge that following a plan driven software development lifecycle is not always feasible and that agile practices can be used in compliance with IEC 62304 without jeopardising the process of achieving regulatory approval.

A number of case studies have been performed in organisations that are actively utilising agile practices in the development of medical device software [12, 32-34]. A common trend emerged out of these case studies that medical device software development organisations recognised it was not possible to fully replace the existing lifecycle with a single agile methodology, but rather they integrated agile practices with the plan driven development lifecycle. Heeager and Nielsen [32] recommend wrapping the traditional lifecycle in a Scrum approach [35]. Likewise, Robres [36] explored the suitability of using Scrum to develop medical device software when attempting to achieve regulatory conformance. In these case, the benefits to be achieved when using agile practices to develop medical device software reflect those achieved when using agile practices in any software development project, they include, reduced development time, reduced costs and increased productivity [37].

4   Proposed Mixed Method SDLC

It is well established that tailoring a SDLC should be done in order to achieve maximum impact [6]. As part of this research we are focusing on integrating agile practices with a plan driven lifecycle. The aim of tailoring a plan driven lifecycle with agile practices is to develop a SDLC which produces the necessary output required for regulatory approval whilst also reaping the benefits associated with utilising agile practices. To perform this tailoring effectively a foundation of a plan driven SDLC was needed. This was required as none of the agile methodologies provide comprehensive coverage of all the necessary development stages required when developing safety critical software [38]. However, Turk [38] does recommended that a safety critical software development project can benefit from combining agile techniques with formal plan driven techniques. The V-Model was chosen as medical device software developers are familiar with the V-Model and it produces the necessary outputs required to achieve regulatory approval and provides guidance for all of the stages of medical device software development. Secondly, research was conducted into which agile practices could be integrated into this plan driven SDLC. The agile practices initially selected were identified through the use of a literature review. The agile practices and SDLC selected must conform to the regulatory requirements whilst still achieving the benefits of tailoring.

4.1 Research Methodology

As part of on-going research into the area of utilising agile practices when developing medical device software a literature review was conducted. This literature was conducted in accordance with Randolph [39]. The literature review began broadly by examining the main software development lifecycles being followed in all software development domains. The literature review then focused on software development in safety critical industries and then concentrated on software development in the medical device domain. Once this part of the literature review was completed, agile methodologies were examined. After this examination was completed the focus moved onto the utilisation of agile practices when developing safety critical software. When conducting the final phase of the literature review a number of seminal papers in the area of using agile practices when developing medical device software were identified [11, 12, 32-34, 40, 41]. In each case study performed, the organisations were initially developing medical device software in accordance with plan driven software development lifecycles. Each of the organisations identified the need to embrace agile software development practices. Consequently upon completion of the implementation of the agile practices each organisation found a noticeable improvement in software quality and increased functionality along with reduced development costs. Table 1 presents summarised findings of the literature review and of the seminal case studies identified. Table 1 shows the case studies identified and which agile practices each of these case studies utilised

4.2 Integrating Agile Practices with a Plan Driven SDLC

As stated when developing a mixed method Software Development Life Cycle a foundation is required. This foundation came in the form of a V-Model for certification proposed by Ge et al. [10]. This V-Model differs from the typical V-Model as it includes development stages for *"Hazard and Risk Analysis"* and *"Regulatory Certification".* These stages are critical which software developed for use in medical devices must complete.

Once the foundation was chosen a method was required to establish which agile practices would be merged with the V-Model. The findings of the literature review were used to establish which agile practices have so far been successfully implemented when developing medical device software. In table 1, a summary of the agile practices suitable for use when developing medical device software are outlined.

	Timeboxed Iterations	Sprint Backlogs	Task Board	Burn Down Chart	Daily Standup Meetings	Self Organising Teams	Continuous Build, Unit Test and Integration	Sprint Retrospective	On Site Customer / Customer Proxie	Test First Development	Pair Programming	Collocated Teams	User Stories
Rasmussen et al.	x				x		x	x					
Vogel	x				x								
Mehrfard et al.													
Spence	x				x	x	x	x	x	x	x	x	
Rottier et al.		x								x			x
Weiguo et al.		x				x							
Heeagers et al.	x	x	x	x	x	x							

Table 1. Case Studies and Agile Practices utilised

Each of the agile practices used in these case studies were then incorporated into the relevant stage of the V-Model. Figure 1 shows the tailored V-Model. This combines the V-Model proposed by Ge et al. and the appropriate agile practices as identified through the literature review.
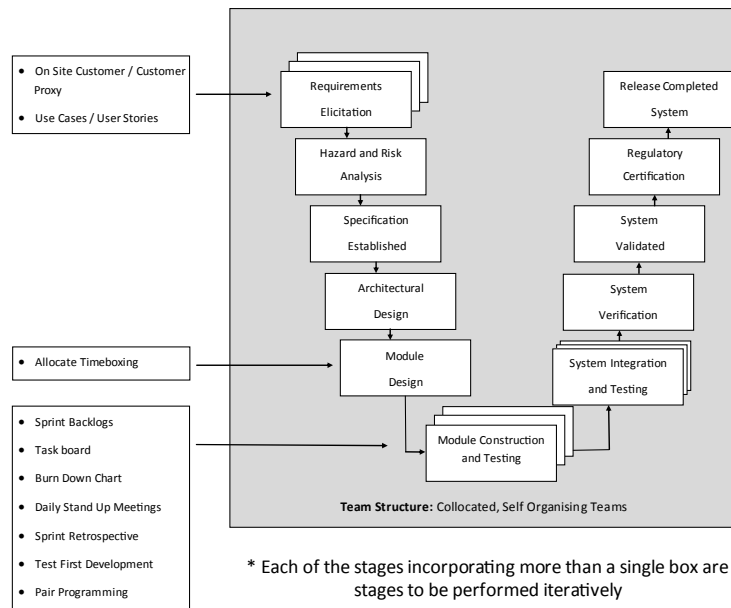
* Each of the stages incorporating more than a single box are
stages to be performed iteratively

Figure 1 V-Model Incorporating Agile Practices

As part of the literature review, agile practices were identified through the key case studies performed. However, none of the key case studies provide information as to which stage of development the agile practices were incorporated. Despite this, we have performed a mapping which matched the agile practices to the appropriate stage of development, as a number of the practices are more suited to specific development stages e.g. Pair Programming = Module Construction and Testing.

5 Conclusions and Future Work

In conclusion, research into instances of agile adoption when developing medical device software has shown that these projects can benefit from integrating agile practices. Medical device software development organisations are bound by regulatory controls. These regulatory controls prevent medical device organisations from wholly embracing a single agile methodology when developing medical device software. The findings of the literature review revealed that there is no SDLC currently available which medical device software developers can follow which produces the necessary deliverables required to achieve regulatory conformance whilst reaping the benefits of utilising agile practices. Research has been conducted in isolation within organisations such as Medtronic and Cochlear, but no evidence exists that suggest the models used in these organisations can be followed with the same level of success in other medical device organisations. A common trend also appeared within each of these case studies. This trend was that no single agile methodology was appropriate for use when developing medical device software and that medical device software development projects benefit from a mixed method approach combining agile practices with a plan driven software development lifecycle. To this end it becomes apparent that medical device software development organisations would greatly benefit from the development of a SDLC that meets both the organisational needs provided from following a plan driven lifecycle and developmental needs from following agile practices.

Research conducted as part of this paper will be used as the foundation for the development of a tailored V-Model which integrates agile practices which are beneficial to the development of medical device software and facilitate the achievement of regulatory compliance. To establish which agile practices should be included into the tailored V-Model, every agile practices within each of the agile methodologies i.e. Scrum, XP, Test Driven Development, Crystal etc. will be examined to determine suitability for use when developing medical device software. Once suitability is determined the appropriate practices will be integrated into the tailored V-Model.

This tailored V-Model will be developed in collaboration with medical device software organisations in Ireland. Once the tailored V-Model is completed it will be thoroughly tested by industry. Once this model has been finalised the objective is to have a medical device software development project fully developed in accordance with the finalised tailored V-Model.

Acknowledgements

Author CVs

Martin Mc Hugh

> Martin received his B.Sc. (Hons.) in Information Technology Management in 2005 and M.Sc. in Computer Science in 2009, from Dundalk Institute of Technology. He is now undertaking research for his Ph.D. in the area of software process improvement for medical devices with emphasis on the usage of agile practices when developing medical device software, as part of the Regulated Software Research Group in Dundalk Institute of Technology.

Fergal Mc Caffery

> Dr Fergal Mc Caffery is the leader of the Regulated Software Research Group in Dundalk Institute of Technology and a member of Lero. He has been awarded Science Foundation Ireland funding through the Stokes Lectureship, Principal Investigator and CSET funding Programmes to research the area of software process improvement for the medical device domain. Additionally, he has received EU FP7 and Enterprise Ireland Commercialisation research funding to improve the effectiveness of embedded software development environments for the medical device industry.

Valentine Casey

> Dr Val Casey is a Senior Researcher with the Regulated Software Research Group in Dundalk Institute of Technology. His previous roles include Senior Lecturer and Research Area Leader at Bournemouth University, Researcher with Lero - the Irish Software Engineering Research Centre at the University of Limerick where he also lectured. He has over 20 years' experience in the software industry. He has also provided consultancy services focusing on software process improvement, software testing and global software development.

Minna Pikkarainen

> Minna Pikkarainen is a Principal Research Scientist in VTT Technical Research Centre of Finland. She has worked in several industrial-driven research projects and project preparations doing close industrial collaboration with large amount of organizations in Europe since 1997. Pikkarainen has been a member of Lero (The Irish Software Engineering Research Centre) during the years 2006-2009. Currently, she is a member of Sirris, The Collective Centre for the Belgian technological industry. Recently, her work and numerous publications have mainly focused on research in the areas of agile development and software innovation.

6 References

1. Cusumano, M. and D. Yoffie, *Software Development on Internet Time.* IEEE Computer, 1999. **32**(10): p. 60-69.
2. Williams, L., *Agile Software Development Methodologies and Practices.* Advances in Computers, 2010. **80**: p. 1-44.
3. Larman, C. and V. Basili, *Iterative an Incremental Development: A Brief History.* Computer, 2003. **36**(6): p. 47-56.
4. http://agilemindstorm.com/agile-glossary/ Accessed 20/02/2012
5. Turk, D. and R. France, *Assumptions Underlying Agile Software Development Processes* Journal of Database Management, 2004. **16**(4): p. 62-87.
6. Conboy, K. and B. Fitzgerald, *Method and developer characteristics for effective agile method tailoring: A study of XP expert opinion.* ACM Trans. Softw. Eng. Methodol, 2010. **20**(1): p. 1-30.
7. Drobka, J., D. Noftz, and R. Raghu, *Piloting XP on Four Mission-Critical Projects.* IEEE Software, 2004. **21**(6): p. 70-75.
8. Greening, J., *Launching XP at a Process-Intensive Company.* IEEE Software, 2001. **18**(6): p. 9.
9. Cawley, O., X. Wang, and I. Richardson, *Lean/Agile Software Development Methodologies in Regulated Environments – State of the Art*, in *Proceedings of Lean Enterprise Software and Systems, .* . 2010, Springer Berlin Heidelberg: Helsinki, Finland. p. 31-36.
10. Ge, X., R.F. Paige, and J.A. McDermid, *An Iterative Approach for Development of Safety-Critical Software and Safety Arguments*, in *Agile 2010.* 2010.
11. Weiguo, L. and F. Xiaomin, *Software Development Practice for FDA-Compliant Medical Devices*, in *International Joint Conference on Computational Sciences and Optimization, 2009. .* 2009: Sanya, Hainan. p. 388 - 390.
12. Vogel, D., *Agile Methods:Most are not ready for prime time in medical device software design and development.* DesignFax Online, 2006(July 2006).

13.   Munassar, N.M.A. and A. Govardhan, *A Comparison Between Five Models Of Software Engineering.* IJCSI International Journal of Computer Science Issues,, 2010. **7**(5): p. 94-101.

14.   European Council, *Directive 2007/47/EC of the European Parliament and of the Council of 5 September 2007*, European Council, Editor. 2007.

15.   AAMI, *ANSI/AAMI/IEC 62304, Medical device Software - Software life cycle processes*. 2006.

16.   European Council, *Commission communication in the framework of the implementation of the Council Directive 93/42/EEC of 14 June 1993 concerning medical devices (Publication of titles and references of harmonised standards under the directive)*. 2010.

17.   FDA, *General Principles of Software Validation: Final Guidance for Industry and FDA Staff*. 2002, *Centre for Devices and Radiological Health*.

18.   FDA, *Federal Register **73** (31), 8637 (15 February 2011)*. 2011.

19.   FDA, *Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices*. 2005, *U.S. Department of Health and Human Services*.

20.   Thompson, B., *FDA Regulations of Mobile Health*, in *mobi Health News*. 2010: Chester Street Publishing. p. 51.

21.   FDA, *Guidance for Industry, FDA Reviewers and Compliance on Off-The-Shelf Software use in Medical Devices*. 1999, *U.S. Department of Health and Human Services*.

22.   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*. 2007, *U.S. Department of Health and Human Services*.

23.   FDA, *General Controls for Medical Devices*. 2009, Food and Drug Administration,.

24.   McCaffery, F., D. McFall, P. Donnelly, and F.G. Wilkie, *Risk Management Process Improvement for the medical device industry*, in *Conference on Software Development (SWDC-REK-2005)* 2005: Iceland. p. 92-103.

25.   Fowler, M. and J. Highsmith, *The Agile Manifesto*, in *Software Development: The Lifecycle starts here*. 2001.

26.   Turner, R. and A. Jain, *Agile Meets CMMI: Culture Clash or Common Cause*, in *1st Agile Universe Conference*. 2002: Chigago.

27.   Boehm, B. and R. Turner, *Using Risk to Balance Agile and Plan-Driven Methods.* IEEE Computer Society, 2003.

28.   McCaffery, F. and V. Casey, *Med-Trace: Traceability Assessment Method for Medical Device Software Development.*, in *European Systems & Software Process Improvement and Innovation Conference, (EuroSPI).* 2011: Roskilde, Denmark.

29.   Bulska, K. and J. Gorski, *Applying Agile Practices to the development of Safety-Critical Software*, in *ICT Young 2011, Scientific Booklets of Faculty of Electronics, Telecommunications and Informatics*. 2011: Gdańsk University of Technology. p. 65-68.

30.   AAMI, *AAMI TIR(SW1)/Ed.1 -- Guidance on the use of agile practices in the development of medical device software*. 2010.

31.   Martin, R.C., *Agile Software Development - Principles, Patterns and Practices*. 2003: Prentice Hall.

32.   Heeager, L.T. and P.A. Nielsen, *Agile Software Development and its Compatibility with a Document-Driven Approach? A Case Study*, in *20th Australasian Conference on Information Systems Compatibility of Agile and Document-Driven Approaches*. 2009: Melbourne.

33.   Rasmussen, R., T. Hughes, J.R. Jenks, and J. Skach, *Adopting Agile in an FDA Regulated Environment*, in *Agile Conference, 2009. AGILE '09.* . 2009: Chicago, IL p. 151-155.

34.   Spence, J.W., *There has to be a better way! [software development]*, in *Agile Conference, 2005. Proceedings* 2005. p. 272 - 278.

35.   Abrahamsson, P., O. Salo, J. Ronkainen, and J. Warsta, *Agile Software Development Methods - Review and Analysis*, in *ESPOO 2002*. 2002, VTT Publications.

36.   Robres, A., *SCRUM in the FDA context.* Agile Record, 2011. **April**(6).

37.   VersionOne, *State of Agile Survey - The Stage of Agile Development*. 2010.

38.   Turk, D., R. France, and B. Rumpe, *Limitations of Agile Software Processes*, in *Third International Conference on Extreme Programming and flexible processes in Software Engineering (XP2002)*. 2002.

39.   Randolph, J., *A Guide to Writing the Dissertation Literature Review.* Practical Assessment Research Evaluation, 2009. **14**(13).

40.   Mehrfard, H. and A. Hamou-Lhadj, *The Impact of Regulatory Compliance on Agile Software Processes with a Focus on the FDA Guidelines for Medical Device Software.* International Journal of Information System Modeling and Design, 2011. **2**(2): p. 67-81.

41.   Rottier, P.A. and V. Rodrigues, *Agile Development in a Medical Device Company*, in *Agile, 2008. AGILE '08. Conference*. 2008.