Medical Device Software Development - A Perspective from a Lean Manufacturing Plant

Oisín Cawley, Ita Richardson, Xiaofeng Wang

Lero - the Irish Software Engineering Research Centre,
University of Limerick
Ireland
{oisin.cawley, ita.richardson, xiaofeng.wang}@lero.ie

Abstract. Developing software for the manufacture of medical devices is a sensitive operation from many perspectives, such as: safety and regulatory compliance. Medical Device companies are required to have a well defined development process in place, which includes software development, and be able to demonstrate that they have followed it through the complete life-cycle of the device. With the increasing complexity of Medical Devices, and more detailed software development regulations among some of the influencing factors, we take a look at how some of these factors have impacted the software development process within a medical device manufacturing plant. We find that tying down your process across the board can have unwanted consequences. As process flexibility is required, we have investigated the usefulness of Lean Software Development.

Keywords: Software Development Process, Medical Device, Regulated Environment, Process Improvement, Lean Software Development

1 Introduction

As would be expected, the concern for human safety takes precedence when it comes to the development of a Medical Device (MD). To ensure the lowest level of risk to safety, various regulatory controls have been established governing the development process. For example within the European Union, the Medical Device Directive [1], [2] define a set of harmonised rules relating to the safety and performance of MDs. One particular aspect which has gained increased attention from a regulatory perspective is the software development life-cycle (SDLC). The software component of a MD is playing an increasingly important role in the construction and operation of MDs, and is becoming more and more complex. This has been reflected in the relatively recent addition to the definition of a MD, for example by the U.S. Food and Drugs Administration (FDA) [3], to include software in its own right as a possible MD.

One thing which is worth noting is that MD software covers embedded-software and also software involved in the manufacturing of the device. It is within this context that we examine the software development process of an Irish based MD

manufacturing plant to see how they configure their internal processes, and what the influencing factors are in achieving regulatory compliance.

1.1 Software for the Medical Device Industry

Medical devices can be defined as being safety-critical. Other domains which fall into this category are the Aviation, Automobile, Railway, and Nuclear among others. As mentioned above, the obvious concern is around safety, and the aim is to minimise the risk of injury to humans to an acceptable level. Each of these safety-critical domains has developed and published standards and guidelines to help achieve the safest possible end-product. For example: the DO-178B guidelines for airborne systems and equipment [4], and the U.S. code of federal regulations title 21 part 820 governing the quality system regulations for medical device manufacturers [5].

Since software is increasingly becoming an integral part of a MD, we have seen an increase in the number of injuries caused to patients which have been directly attributed to the software component. FDA analysis of 3,140 medical device recalls between 1992 and 1998 found that 242 (over 7%) were attributable to software [6]. Significantly, of the recalls in 2007 of what the FDA classify as life-threatening, 23 of them involved faulty software [7]. As a consequence, regulatory controls are continuously being reviewed and adapted to this ever advancing technological landscape. In 2006 an international standard (ANSI/AAMI/IEC 62304:2006) was published which governs the MD SDLC processes [8]. Now widely adopted, IEC 62304 establishes a common framework for medical device life-cycle process by describing a set of processes, activities, and tasks that are required within a MD SDLC.

However, reading the standards can lead to thinking that a waterfall-type software development methodology is what will best meet the requirements. This is in fact not the case, and Annex B of the IEC 62304 standard specifically clarifies that: "This standard does not require a particular SOFTWARE DEVELOPMENT LIFE CYCLE MODEL". This allows companies to employ whatever methodology they prefer, for example, Incremental or Evolutionary. Typically MD companies employ a traditional SDLC model (waterfall or V), but lately, more focus is being given to examining how these companies can improve their SDLC processes, for example by employing a more iterative development methodology [9], [10].

1.2 Lean Software Development

The concept of Lean Software Development can be thought of as the merging of the principles of Lean Manufacturing [11], [12], with software development practices. Lean's primary focus is on the identification and elimination of waste from the process. Waste being defined as "any human activity which absorbs resources but creates no *value*" [13]. So lean thinking "is lean because it provides a way to do more and more with less and less – less human effort, less equipment, less time, and less space – while coming closer and closer to providing customers with exactly what they want" [13].

Eliminating waste, when translated into software engineering terms, can mean the elimination of defects (bugs) in the code. This may seem an obvious goal of

developing software, but the creation of formal mechanisms to achieve this began to show the power of doing this in a systematic fashion. This is one of the cornerstones of what Lean Software Development is founded on, finding and fixing defects early in the development process. Many of the Agile software development practices [14], [15] can be seen as supportive of a lean philosophy (Fig. 1). For example, the agile practice of test driven development (TDD) [16] in order to find defects early by continuous testing and thus reducing the cost of rework later. Many more such practices have been mapped by [17] and [18].

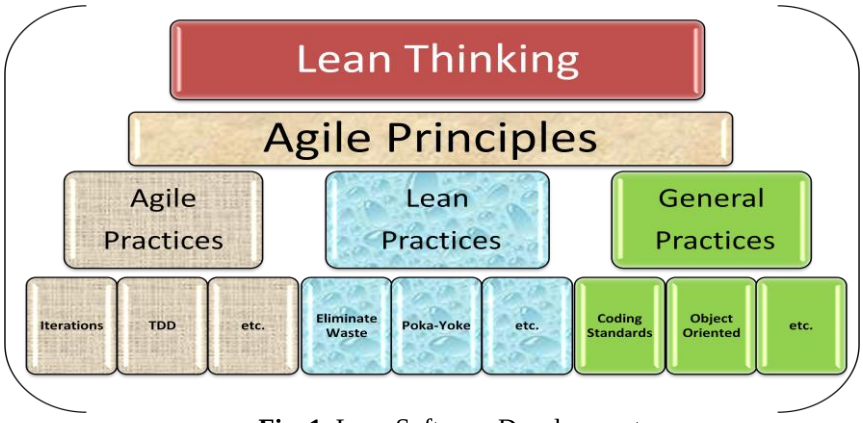

Fig. 1. Lean Software Development

Of interest therefore is how such 'lean practices' can be utilised in the MD domain, something the industry is also looking at [10]. With a focus on an iterative approach to software delivery, and favouring less rather than more documentation [19], companies can shy away from such practices out of fear of a costly non-compliance outcome. Slowly, however, we are seeing more and more reports from companies who are trialling lean approaches in this domain [20], [21], [9].

2 Research Approach

Following on from a systematic literature review of software development within safety-critical regulated environments [22], we were interested in investigating further how the various regulations and other influencing factors affect the software development process within the MD domain, and how a lean perspective could be beneficial. Our approach to this was to undertake a case study within a MD company, using [23], [24] as guides. Taking an interpretative approach, we requested one-to-one interviews with a cross section of the organisation involved in projects which had a software development component. Eight onsite interviews were performed lasting between sixty and ninety minutes each. The interviewees all had relevant firsthand experience of the complete product development process and the governing policies and procedures. They included senior software developers, a senior quality engineer, a process engineer, and project leaders. Their work experience ranged from 7 to 18 years, and from 3 to 9 years within a MD context.

With the permission of the interviewees, the interviews were recorded and later transcribed and analysed using qualitative data analysis techniques such as open and axial coding as described by [25]. Other artefacts were also gathered while on site, such as documents describing internal processes and procedures, organisational charts, project metrics, corporate policies, standards, presentations, and email correspondence. Together with on site observations, all these artefacts were used within the case study to gain a more holistic view of the working environment.

2.1 The Company

MedTech (a pseudonym) is a large US medical device company with manufacturing facilities located in the United Sates and Ireland. Within the particular plant we investigated, the MDs do not have any embedded software, but a large effort is required in developing and maintaining the automation software necessary for manufacturing the devices. The plant performs a combination of research and development as well as commercial MD manufacturing.

Like most businesses, the current global economic conditions have also taken their toll on MedTech. They went through a process of workforce reduction in recent years, and during our research we noted how this reduction has affected the way the employees work. As a consequence the daily endeavours of how they comply with the regulations, has been brought into the spotlight, something we will discuss below.

As a committed Lean organisation, MedTech maintains quite an impressive visual display of their values, lean initiatives and achievements. What was interesting from the interviews therefore were the responses to questions probing software development process improvement from a lean point of view.

3 Research Findings

From our case study analysis, we show the key components which have shaped the development process within MedTech (Fig. 2).

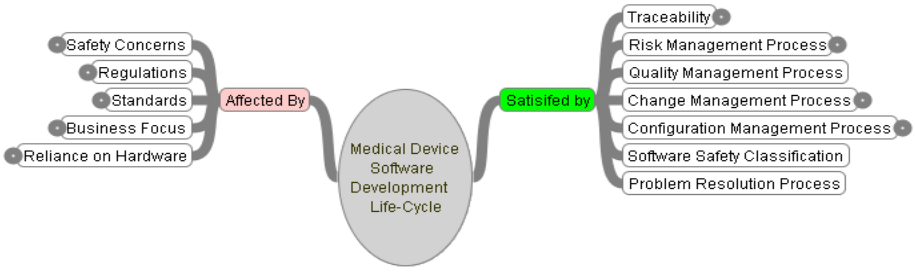

Fig. 2. Overview of key influences on the SDLC within MedTech

Due to space limitations we expand here on just two of the important factors which emerged from the case study, namely: Regulations and Business Focus. These factors hugely influenced how the SDLC evolved to what it is today, and continue to exert pressure on various processes within the organisation, but particularly the product development process, of which the SDLC is a key sub-process.

3.1 Regulations

Ensuring that the SDLC meets the requirements of the regulations typically means that internal processes are defined and documented and have been mapped to a relevant standard. For example, an internal risk management process may be mapped to the international standard ISO 14971:2007 (international standard for the application of risk management to medical devices). However, these standards are open to interpretation and even with the aid of guidance documents such as those published by the FDA's CDRH (Centre for Devices and Radiological Health), and also due to the different classifications of MDs, it can be difficult to know exactly what the auditors expect. As a result there may be a tendency to overdo it in terms of the process, in a 'just to be sure' approach.

MedTec's growth was partly due to the acquisition and amalgamation of smaller companies. Each of these had their own internal processes, and little attention was given to the actual software development process. As stated during an interview: "*The development bit in the middle there was a bit of a black art*". However following an Audit in 2005, a deficiency was found, in that they were unable to show traceability back to a user requirement for a particular controlling software parameter. The serious nature of this spurred them into a corporate initiative to revamp their processes. A corporate wide SDLC was defined governing all aspects of software development and which mapped to the relevant regulations.

This corporate umbrella-policy was designed to cover all relevant activities: system upgrades, new product design, and production issue resolution. It was therefore necessarily high level, and consequently another level of granularity was required to govern at a site level. The site level SDLC process they implemented mapped to the corporate policy and was therefore considered to be in compliance with the regulations. The various teams trust that these policies are regulatory compliant. When asked specifically about IEC 62304 and how it affects their processes, a response was: "*We've got a regulatory group that assess standards against corporate policy...So I assume that activity would have happened up there*".

But what is evident from the case study is that, three or so years on, there is widespread discontent with the process. This has been acknowledged at a corporate level, and they have kicked off an initiative to review it. The huge amounts of documentation, the number and level of approvals, the time required, the cost, were the types of issues the interviewees had with the process. The following quote summarises the mood: "*You can do all this [the process] and deliver [bad quality] to your customer ... which is why it's seen as a failure in the overall organisation. Project managers and R&D complain about it, engineering managers complain about it, nobody understands it, it's just a noose, it's just a pain really*".

What we have observed is that, by means of an initiative to improve the software development process, some very undesirable side effects have occurred. One of the reasons for this is that the process is indiscriminate of the focus of the activity being carried out. Within the R&D projects for example, the nature of the work is that they do not know up front exactly what it is they want the software to do. A rough idea of the requirements is known but many of the intricacies, for example, what margin of error is required, will only be known once the software and equipment have been prepared and trialled. So while the process calls for full requirements disclosure up

front, followed by a lengthy change control process for any subsequent changes, this does not lend itself well to an iterative-type product development process typical of most R&D activities. Of one project, the following was said of the software process, "*... [it would] add 4 months onto the schedule of delivering the equipment to the customer*". Such considerable overhead can make people look for short cuts, and that can't be seen as a good thing.

3.2 Business Focus

Breaking down the components of the business focus category we see the main influencers as illustrated in Fig. 3:

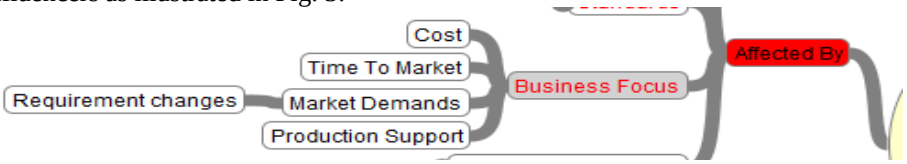

Fig. 3. Component elements of the Business Focus factor within MedTech

Again due to space restrictions, we describe two of these, Cost and Time-To-Market which were the highest ranked in terms of the coding process, and briefly discuss how each one affects the software development process.

3.2.1 Cost

Developing Medical devices is an expensive business, both in terms of the opportunity cost (time to market, which we present in section 4.2.2) and the cash burning from simply running the whole process. In a recent U.S. economic report [26], the premium paid to employees in the Medical Technology Industry (MTI) is highlighted: "One of the outstanding characteristics of the medical technology industry is the strong pay scale ... almost a 40 percent premium for jobs".

MedTech also went through a process of headcount reduction to reduce costs. As a result some process improvement initiatives have suffered by being either cancelled or postponed. Another effect has been that product managers are less willing to pay for software development unless it is really necessary. Because MedTech operate an internal cross-charging process, managers are less likely to proceed with software or equipment enhancements once they see the project estimate. Thus the effect on the developers is that they spend a significant amount of time producing estimates for software development which might never proceed. From a lean perspective this type of task-switching introduces waste into the process and should be minimised: "Every time software developers switch between tasks, a significant switching time is incurred as they get their thoughts gathered and get into the flow of the new task" [17].

MedTech also utilises contract resources on an as-needed basis which can be hugely expensive "*We pay contract validation resources to do all this work, to the scale of millions per year*". However the duration of the software validation cycle can be quite difficult to estimate, and so when contract resources are employed, costs can very quickly accumulate. To paraphrase from one of the interviewees: if one were a

sceptic, one might say it is in the interest of the validation contractors to find issues, which will add further validation needs. The problem within the MD domain is that validation needs to meet the requirements of the regulations, which, as the FDA themselves acknowledge, is difficult to define precisely [6].

3.2.2 Time To Market

Getting a new product through development as quickly as possible in order to move into the clinical trials and then commercialisation phases is becoming ever more critical to MD companies. Once a launch date has been set, there is very little appetite for delays. As one interviewee put it "*time is generally the biggest priority. Time out weighs cost a lot of the time here*". This has a consequence for the software development process in that there is little tolerance for changes to process that might introduce risk to compliance and therefore affect time to market. Following one particular process improvement initiative to automate a production line which led to a delay in time-to-market, one interviewee reported "*... the tolerance has gone to zero now for equipment upgrade or equipment strategy causing a delay in product time to market*".

The concentration on time to market has also created organisational inertia to change within the new product introduction process. This may be typical within a MD environment, [27] states: "In such a world, any significant change to culture or process can be difficult. Sometimes it is difficult because of the inertia inherent in a large organization". During one interview it was suggested that they could possible improve their manufacturing process if they spent more time researching newer technologies and techniques "*Time to market is the big deal here ... It's very rare that we get to develop prototype equipment, [and] test it*". This includes the SDLC, because they interact with the development group and would see an iterative approach being more effective for them as opposed to the 'waterfall' type approach which their process requires. What has therefore happened is they have found ways to minimise the process overhead, bending the rules of the process along the way.

4. Discussion

Our investigations have brought to light some interesting effects that certain factors, such as regulations, have on the software development process within a MD plant. A possible outcome of having to adhere to regulatory requirements can be the corporate enforcement of a process. Speaking from the perspective of a MD firm, [27] states: "In our safety-critical world, we strongly believe that a robust process is an important element to insuring high quality. The side effect of that strong belief can be an over-reliance on prescriptive, mandated process rules that take the approach of imposing discipline upon a team". Although it is good to have a well defined process, it seems that it is easy to over shoot the mark. A process which is not flexible enough to allow for all aspects of the business, for example the R&D section which requires an iterative type of process, will result in frustrated employees and possibly short cutting of processes.

4.1 Leaning The Software Development Process

It is important to have a work ethic of continuous improvement within an organisation. Many process improvement models such as CMMI (level 5) [28], ISO/IEC 15504-5 [29] refer to the highest level of process maturity having continuous improvement as a core element. This is similar to the lean principle of striving for *perfection* [13]. This ethos is evident within MedTech, which has been proactive in tackling the issues by various process improvement initiatives. For example, conscious of the issue of task switching when quotes are needed for potential projects, a process MedTech introduced that helps reduce this, is the availability of an internal web based tool. The tool poses a number of key questions and then gives a high level estimate of the project cost, "*... the number could be out by 20 or 30 % but it's just to give people an order of magnitude*". This allows managers to gauge whether or not to pursue the project further, thus reducing the number of non-value-add project estimates the developers get involved in.

The same approach can also work with the software developers. One of the theses presented by [30] based on their experience of software process improvement initiatives is that "Developers are motivated for change; if possible, start bottom-up with concrete initiatives." There are various tasks the MedTech developers have identified that could lead to process efficiencies, such as better code re-use, code reviews, use of testing tools, and skills development. Many such tasks are independent of the regulatory environment they operate in, however what is needed is some form of organisational support to find time for employees to execute on these initiatives. Within the lean manufacturing world, this is achieved via policy deployment [13] pg. 95: "The idea is for top management to agree on a few simple goals for transitioning [and] to designate the people and resources for getting the projects done". Some software development changes however, could be problematic in a MD company, for example the practice of refactoring source code [31] may cause unexpected results to previously completed (and possibly certified) code [32], [33]. A certain amount of caution is required.

Hiring contract staff is not an uncommon approach to manage surges in capacity, and with Global Software Development [34] becoming common place in many large companies, one could imagine that eventually it will become more common within the MD domain also. The case study has shown that cost considerations are less important than in other industries, especially when it comes to new product R&D. However the long project timelines typical of a new MD generate very large costs, something the MD industry is not immune to. In tackling the huge expense of hiring contract validation engineers, two lean process changes helped to reduce this cost exposure in MedTech. Firstly in relation to the sourcing of validation engineers: "... *we've reduced the numbers from a contractor base and they're not as pernickety now*". Secondly, taking what can be seen as a lean approach to contract management [17], they changed the contract terms from time and materials to fixed price, and any additional work would have to be renegotiated with the requesting manager who would be far more reluctant to incur additional project costs: "*Now they're very effective at managing scope*".

A big question researchers should be asking is how to apply software process improvement within this type of domain while keeping risk to compliance as close as

possible to zero. Some work has been done within embedded-software domains on how to choose the most appropriate methodology [35] and how agile practices should be considered [36]. However, the only model we found relating to mission or life-critical agile adoption [37] describes a stepped approach to deciding which agile practices are suitable depending on the system's characteristics and qualities. While their framework is aimed at any mission or life-critical system, it does not go into much detail about the various regulatory requirements.

Within MedTech there is a great work ethos of process improvement, and within the software development group this is no different. There is acknowledgement that they could be doing things better "*Everyone using this [the process] is quite frustrated by how much time we spend having to document and test stuff .... And it costs a lot of money*". They are actively engaged in analysing their current process in order to "*make [it] more effective, efficient and more lean than what we do now*". This is easier said than done, and especially within the SDLC.

Lean, having its origins in a manufacturing setting, has had limited success in penetrating areas such as software development. However, using the concept of lean software development we can see that MedTech has instigated some practices that have made some initial process improvements. Information hiding through modularisation and componentisation which remove complexity, parameterisation making modules less implementation specific, and code re-use are suggested lean practices [17]. MedTech seem to be going in the right direction having begun to build a library of software objects and re-usable components. As one interviewee attested "*From the testing side I find it personally a lot better because I know that that piece of code, I don't have to check it, it's been done already and working reliably*".

They have also reduced waste in terms of communication between the software group and the more equipment focused group. Following a recent re-organisation it was reported that "*those two groups have merged under the one manager an there's been a lot more interaction*". Also, the developers themselves have embarked on their own initiative to cross-train each other so as to increase the level of expertise within the group "*Now we interchange rolls so those guys sometimes they do machines and sometimes ... or whoever will do databases, screens, something like that*".

Further possibilities will be worth exploring for MedTech, such as establishing a TDD approach [20]. With their hardware dependencies, TDD, combined with the use of testing techniques which decouple the hardware such as bracketing-out, mocks, and stubs [38], and hardware simulators [39] could offer process improvements well worth pursuing.

5 Conclusion and Future Work

Having examined the SDLC within a MD manufacturing plant, we identified a number of key influencing factors (Fig. 2) and expanded on two in this paper, namely regulations and business focus. While many MD companies are pre-occupied with achieving regulatory compliance, we have seen in this example how it can lead to a feeling of over doing it by applying a heavy process to all aspects of the software life-cycle. We also identified some of the key business drivers (Fig. 3) and again expanded on just two of these: cost and time-to-market. The MD industry appears to

becoming much more competitive and cost focused, and therefore companies are looking at ways to improve their processes but without affecting regulatory compliance. Since there does not seem to be a mechanism for quantifying just how much process is enough, it would be beneficial for a focused assessment of existing processes which could indicate where too much rigour is being applied and therefore the possibility to reduce the amount of work required.

Lean software development, although still not very well defined, offers the potential for transferring the principle of lean manufacturing into software development and thereby achieving some of the benefits seen by lean initiatives in other parts of the organisation. What would be useful is a reference model of lean software practices which can be employed while not affecting regulatory compliance.

The MD domain is seen as an area with huge growth potential particularly relevant within Ireland [40]. As an Irish based software engineering research centre, we therefore find it very relevant to conduct research into this domain. Indeed the outlook for this domain, in a global sense, is one of great advancements in technology leading to smaller, more complex devices merged with physiological, biological, engineered and physical systems. Importantly, it is anticipated that the current software development methodologies for such nanoscale systems will have to fundamentally change [41]. The challenge for the research community, therefore, is to develop architectures and methodologies appropriate to supporting these advancements while keeping an open mind as to how the regulatory bodies are likely to respond. Consequently, any process improvement, assessment, or reference model should be future proofed to allow for and support such innovations.

Acknowledgements:
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 03/CE2/I303_1.

5 References

1. EU, Council Directive 93/42/EEC of the European Parliment and of the Council, Concerning Medical Devices, E. Council, Editor 1993, Official Journal of the European Union.
2. EU, Directive 2007/47/EC of the European Parliment and of the Council, 2007, Official Journal of the European Union.
3. FDA. U.S. Food and Drugs Administration. Available from: (www.fda.gov). Accessed 10 March 2011.
4. RTCA, DO-178B, Software Considerations in Airborne Systems and Equipment Certification, January 1992, RTCA (Radio Technical Commission for Aeronautics).
5. FDA, Code of Federal Regulations 21 CFR Part 820, U.F.a.D. Administration, Editor April 2009.
6. CDRH, General Principles of Software Validation; Final Guidance for Industry and FDA Staff, 2002, FDA.

7. IEEE, IEEE Reliability Society - Annual Technical Report 2008.
8. ANSI/AAMI/IEC, 62304:2006 Medical Device Software-Software life cycle processes, 2006, Association for the Advancement of Medical Instrumentation. p. 67.
9. Spence, J.W. There has to be a better way! In AGILE Conference 2005, July 24 - July 29, 2005. Denver, CO, United states: Inst. of Elec. and Elec. Eng. Computer Society.
10. Bosch, T. Medical Device Software Development—Going Agile, www.mddionline.com/article/medical-device-software-developmentmdashgoing-agile. Accessed 10 March 2011
11. Womack, J.P., D.T. Jones, and D. Roos, The Machine That Changed The World: How lean production revolutionized the global car wars. 2007: Simon & Schuster.
12. Liker, J., The Toyota Way 2003: McGraw-Hill.
13. Womack, J.P. and D.T. Jones, Lean Thinking : Banish Waste and Create Wealth in Your Corporation 1996: Simon & Schuster.
14. Cockburn, A., Agile Software Development. The Agile Software Development Series, ed. C.a. Highsmith, 2002.
15. Highsmith, J., Agile Software Development Ecosystems, 2002: Addison Wesley.
16. Beck, K., Test Driven Development: By Example, 2002: Addison-Wesley Professional.
17. Poppendieck, M. and T. Poppendieck, Lean Software Development: An Agile Toolkit, 2003: Addison-Wesley Professional.
18. Hibbs, C., S.C. Jewett, and M. Sullivan, The Art of Lean Software Development, 2009: O'Reilly Media.
19. AgileAlliance. Manifesto for Agile Software Development, Available from: www.agilemanifesto.org/. Accessed 10 March 2011.
20. Rottier, P.A. and V. Rodrigues. Agile Development in a Medical Device Company. In Agile, 2008. AGILE '08. Conference.
21. Lin, W. and X. Fan. Software development practice for FDA-compliant medical devices. In 2009 International Joint Conference on Computational Sciences and Optimization, CSO 2009, April 24, 2009 - April 26, 2009. Sanya, Hainan, China: IEEE Computer Society.
22. 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, p. 31-36. Helsinki, Finland, October 2010: Springer Berlin Heidelberg.
23. Yin, R.K., Case Study Research: Design and Methods Paperback ed. Applied Social Research Methods. 2009: Sage Publications, Inc.
24. Miles, M.B. and M.A. Huberman, Qualitative Data Analysis: An Expanded Sourcebook (2nd Edition)1994: Sage Publications, Inc; 2nd edition.
25. Strauss, A. and J. Corbin, Basics of qualitative research: grounded theory procedures and techniques.1990: Sage Publications.
26. The Lewin Group, Inc., State Economic Impact of the Medical Technology Industry, 2010. http://www.lifechanginginnovation.org/. Accessed 10 March 2011.
27. Weyrauch, K. What are we arguing about? A framework for defining agile in our organization. In Agile Conference, 2006.
28. CMMI, Capability Maturity Model Integration V1.3, http://www.sei.cmu.edu/cmmi. Accessed 10 March 2011.
29. ISO/IEC 15504-5: 2006 Information Technology – Process Assessment – Part 5 : An exemplar Process assessment model, 2006.
30. Conradi, H. and A. Fuggetta, Improving software process improvement. Software, IEEE, 2002. 19(4): p. 92-99.

31. Fowler, M., Refactoring-Improving The Design Of Existing Code. Hardback ed. Object Technology Series, ed. Booch, Jacobson, and Rumbaugh. 2000: Addison-Wesley Professional.
32. Chisholm, R.A., Agile Software Development Methods and DO-178B Certification, In Division of Graduate Studies and Research. 2007, Royal Military College of Canada.
33. Ronkainen, J. and P. Abrahamsson. Software development under stringent hardware constraints: do agile methods have a chance? In Extreme Programming and Agile Processes in Software Engineering. 4th International Conference, XP 2003. Proceedings, 25-29 May 2003. Berlin, Germany: Springer-Verlag.
34. Herbsleb, J.D. and D. Moitra, Global Software Development. IEEE Software, 2001. 18(2): p. 16-20.
35. Kettunen, P. and M. Laanti, How to steer an embedded software project: tactics for selecting the software process model. Information and Software Technology, 2005. 47. IEEE: p. 587-608.
36. Srinivasan, J., R. Dobrin, and K. Lundqvist. 'State of the Art' in Using Agile Methods for Embedded Systems Development. In Computer Software and Applications Conference, 2009. COMPSAC '09. 33rd Annual IEEE International.
37. Sidky, A. and J. Arthur, Determining the Applicability of Agile Practices to Mission and Life-Critical Systems, In Proceedings of the 31st IEEE Software Engineering Workshop 2007, IEEE Computer Society.
38. Van Schooenderwoert, N. and R. Morsicato. Taming the embedded tiger - Agile test techniques for embedded software. In Proceedings of the Agile Development Conference, ADC 2004, June 22 - June 26, 2004. Salt Lake City, UT, United states: IEEE Computer Society.
39. Mueller, G. and J. Borzuchowski, Extreme embedded a report from the front line. In OOPSLA 2002 Practitioners Reports 2002, ACM: Seattle, Washington.
40. Expert-Group-on-Future-Skills-Needs, Future Skills Needs of the Irish Medical Devices Sector. 2008, www.skillsireland.ie/publication. Accessed 10 March 2011.
41. High Confidence Software and Systems Coordinating Group, High-Confidence Medical Devices: Cyber-Physical Systems for 21st Century Health Care - A Research and Development Needs Report, 2009, The Federal Networking and Information Technology Research and Development (NITRD) Program.