

# Technology Engineering for Medical Devices - A Lean Manufacturing Plant Viewpoint

Sai Sruthi Gadde<sup>1</sup>, Venkata Dinesh Reddy Kalli<sup>2</sup>

Ph.D. in Information Technology, University of Cumberlands, Louisville, Kentucky, USA<sup>1</sup>

Research Scientist, Cardiac and Vascular Group, Medtronic, Tempe, USA<sup>2</sup>

**Abstract:** The development of software for medical device manufacturing is a delicate process from many points of view, including protection and compliance with regulations. Medical device companies must be able to develop and show that they have carried out the development process over the entire life cycle of the product. Technology development included. As medical devices are increasingly sophisticated and software development regulations are some of the causes, we discuss how both factors have influenced the process of software development in a medical device manufacturing plant. We think the approach may have negative implications around the board. We also explored the utility of Lean Software Development because process consistency is required.

**Keywords:** Medical Device Software, FDA, SDLC, Lean Software Development

## I. INTRODUCTION

As expected, human health issues take priority when it comes to designing a Medical Device (M.D.). Several regulatory controls regulating the production process have been developed to ensure the lowest level of risk to health. For example, the Medical Devices Directive [1],[2] establishes a collection of harmonized rules for the protection and efficiency of M.D.s within the European Union. One specific topic that has received more considerable attention from a regulatory point of view is the life- of software development (SDLC). An MD's software component plays an increasingly important role in building and running M.D.'s and is becoming increasingly complex. That was expressed in the relatively recent introduction, for example, by the U.S., to the concept of an M.D. Food and Drugs Administration (FDA) [3], to integrate software as a potential M.D. into its right. One thing worth noting is that M.D. software includes embedded software as well as software that is involved in computer manufacturing. It is in this context that we analyze an Irish-based manufacturing plant's software development process to see how they configure their internal processes, and what contributing factors are in achieving regulatory compliance.

## II. SOFTWARE FOR THE MEDICAL DEVICE INDUSTRY

Medical devices can be described as being essential to health. Many fields that fall under this category include Aircraft, Car, Railway, and Nuclear. As stated above, the apparent concern is about health, and the goal is to mitigate to an appropriate level the risk of injury to humans. Each of these critical safety areas has produced and published standards and guidelines to help provide the safest end-product possible. For example, the DO-178B guidelines for airborne systems and equipment [4], and the U.S. federal regulatory code title 21 section 820 regulating the quality control regulations for manufacturers of medical devices [5].

As software is becoming an integral part of an M.D. increasingly, we have seen a rise in the number of injuries caused to patients directly due to the software portion. Between 1992 and 1998, the FDA review of 3,140 medical device recalls found that 242 (over 7 percent) were software attributable [6]. Of the 2007 reminds of what the FDA classifies as life-threatening, substantially 23 of them involved defective software [7]. As a result, regulatory frameworks are being regularly checked and adjusted to this ever-advancing technical environment. In 2006 an international standard was published (ANSI / AAMI / IEC 62304:2006), which governs the processes of the MD SDLC [8]. Now widely adopted, IEC 62304 provides a standard structure for the life-cycle phase of medical devices by defining a collection of processes, activities, and tasks required through an MD SDLC.

Nonetheless, reading the specifications can lead to thinking that what best suits the requirements is a waterfall-type software development methodology. In addition, this is not the case, and Annex B of the IEC 62304 standard specifies that: –This standard does not include a particular SOFTWARE DEVELOPMENT LIFE CYCLE MODEL. It helps organizations to use whatever approach, such as Incremental or Evolutionary, they choose. M.D. companies usually use a conventional SDLC model (waterfall or V), but recently more attention has been given to examining how these companies can enhance their SDLC processes, for example, by the use of a more iterative method of production [9], [10].

### III. LEAN SOFTWARE DEVELOPMENT

The definition of Lean Software Development can be seen as combining the Lean Manufacturing principles [11],[12] with software development practices. The primary focus of Lean is on finding and disposing of waste from the operation. Waste is defined as any human activity that consumes resources but does not generate any value [13]. And lean thinking – is lean as it offers a way to achieve more and more with less and less resources – less human energy, less machinery, less time and less space – while moving closer and closer to providing consumers with exactly what they want [13].

When translated into software engineering terminology, eliminating waste can mean removing defects (bugs) in the code. This may seem an apparent aim of software development, but the emergence of structured structures to accomplish this started to display the power to do this in a systematic fashion. This is one of the cornerstones of what Lean Software Development is focused on, identifying and addressing shortcomings early in the process of development. Many of the Agile software development practices [14],

[15] Therefore, it is of concern if these upper-class methods can be used in the M.D. domain, something that the industry often looks at [10]. Focusing on an iterative approach to the delivery of software, and preferring less than more documentation [19], businesses may shy away from these activities out of fear of expensive non-compliance outcomes. Little by little, however, we are seeing more and more reports from companies' trialing lean solutions in this area [20], [21], [9].

### IV. RESEARCH APPROACH

Following a thorough analysis of software development in secure, regulating environments [22], we were interested in exploring further the effect on software development in the M.D. domain of the different regulations and other contributing factors and the usefulness of the lighting perspective. Our approach to this was to conduct a case study within an M.D. company, using [23],[24] as guides. We called for one-to-one interviews, using an interpretative approach, with a cross-section of the organization interested in projects that had an aspect of software development. Eleven on-site interviews lasting 50 to 80 minutes each were performed. Everybody was thoroughly familiar with the whole process and the policies and procedures that control product development. Senior software developers, senior production engineers, process engineers, and team members were among those responsible. The MD's history requires 8 to 17 years of professional experience and 4 to 8 years of research in the interviews, the qualitative data processing techniques such as open and axial code, as described in [25], were documents and later transcribed and analyzed by permission of the interviewer. Other items were also gathered while on-site, including records outlining corporate procedures, organizational diagrams, mission updates, business plans, recommendations, presentations, and email correspondence. Both these objects and observations on site were used for a practical view of the work environment in the case study.

### V. THE COMPANY

MedTech (a pseudonym) is a major US medical device corporation based in the United States and Ireland with manufacturing facilities. The M.D.s do not have any embedded software at the specific plant that we examined. Still, a great deal of effort is required to create and maintain the automation software needed for manufacturing devices. The plant carries out a mix of R&D and commercial M.D. production. Unlike other businesses, MedTech also has its toll on the existing global economic conditions. In recent years, they have gone through a cycle of reducing jobs. Through our study, we have seen how this decline has impacted jobs. MedTech, as a dedicated Lean company, retains its ideals, its lean programs, and its successes in quite a remarkable way. The answers to questions about developing software creation from a lean point of view were also interesting from the interviews.

### VI. RESEARCH FINDINGS

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

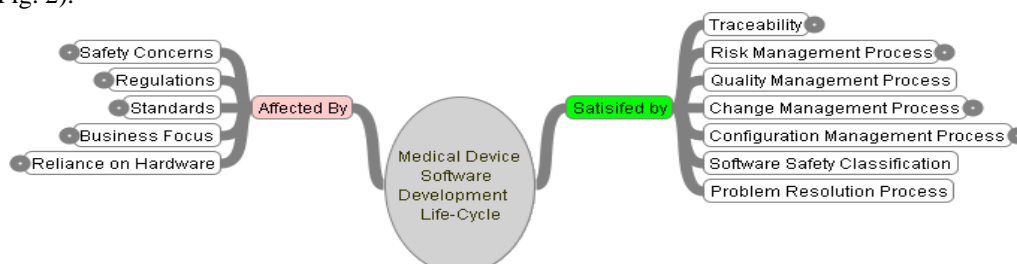


Fig. 2. Overview of critical influences on the SDLC within MedTech



Due to space constraints, we elaborate here on only two major factors, i.e., legislation and market emphasis, which emerged from the case study. Such factors have significantly influenced the way that the SDLC has developed to the present level and continue to exercise pressure on several processes within the organization, but mainly on the development process, the SDLC being the key subprocess.

## VII.REGULATIONS

Making sure that SDLC complies with regulatory requirements usually involves identifying and recording internal processes and mapping them to the correct level. In the context of an international standard for risk management applications in medical devices, an external risk management framework can be compared to the ISO 14971:2007 (international standard). Such criteria are, however, subject to interpretation, and because of the various classifications of M.D.s, it can be difficult to precisely know what the auditors expect from the FDA's CDRH (Center for Devices and Radiological Salvation). In effect, there may be a tendency to overdo the process in a way that's secure.

MedTech's success was due in part to smaller businesses being purchased and fused. Everyone had its own internal procedures, and the actual software development process was given little attention to it. As an interview said, "There was a bit of a black art about creation in the south. Nonetheless, a weakness was noticed after an audit in 2005, because the traceability of a specific program control parameter was incapable of demonstrating a user requirement. The gravity of this led them to a business effort to reform their processes. A broad SDLC for all aspects of software development was specified and mapped accordingly.

This company strategy has been designed to include all related activities: improvements to the program, new product design, and the resolution of development issues. Consequently, it was inevitably high, and a further granularity was required to rule at a site level. The SDLC process was carried out at the site level and was therefore seen as in keeping with the corporate policy. The different teams are assured that the regulatory policies follow. When explicitly asked about IEC 62304 and its mechanism, the response was: "We have a regulatory group which evaluates corporate policy standards.

But the case study indicates that there is general skepticism about the mechanism three or so years later. This was noted at the corporate level, and an effort to study it was initiated. The vast volumes of paperwork, the number, and extent of approvals, the amount of time and expense were the issues that the interviewed engaged in the process. The following quote sums up the mood: "You can all do this [process] and provide your customer with [bad quality] ... That's why the overall enterprise is seen as a failure. Project managers and research and development executives complain, engineers complain about it; nobody knows it, it is just a noose, I don't get it.

We have found that some very unexpected side effects have arisen in an effort to enhance the software development process. One of the reasons for this is that the focus of the action is indiscriminate. The essence of research and development ventures is, for example, that they are not necessarily sure of what applications they want to do. A rough understanding of the specifications is understood, but many of the intricacies, such as increasing error margin is required, are not understood until the software and devices are created and tested. Although the process calls for complete disclosure of specifications beforehand, followed by a lengthy process of change management for subsequent changes, this does not suit an iterative software development process typical of the most research and development activities. The following about the software process has been said of one phase, — [It would] add four months to the customer regular delivery schedule. Having a high overhead will cause people to look for short cuts, and that cannot be viewed as a positive thing.

### Business Focus

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



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

Because of space constraints again, we identify two of them, the most important ones for the coding process, cost and time-to-market, and briefly address their impact on the process of software development.

### Cost

Medical equipment manufacturing is a costly business in both the price of opportunity (time to market as we present in section 4.2.2) and cash usage by merely running the entire operation. A recent U.S. economic report [26] highlights the premium payable to Medical Technology (MTI) staff: "The big wage scale is one of the outstanding features of the

medical technologies industry ... A bonus for workers nearly 40 percent GENERAL.

MedTech has also undergone a phase of reduction of headcount costs. As a consequence, programs to strengthen procedures have been scrapped or deferred. It has also been a consequence that product managers are less likely to pay for the production of software when it is needed. Since MedTech is running an internal cross-loading cycle, managers are less likely to continue upgrading software or equipment once they see the project estimate. Therefore, the effect is that developers waste significant time generating forecasts for the development of applications that will never take place. This kind of task transition leads to drain from a lean perspective and should be minimized: As software developers turn between tasks, a significant time is expended as they gather their ideas and join the process of new job [17].

MedTech also uses as-needed contract tools, which can be very costly. They spend millions each year to verify the contract for all of this work. But it can be very difficult to predict the length of the software validation process, and so costs can very rapidly escalate when contract services are used. In a paraphrase of one interviewee: if one is suspicious, the validation contractors would have an interest in discovering problems that would create more needs for validation. The challenge in the M.D. domain is that certification must comply with the regulations that are approved by the FDA itself [6].

### **Time To Market**

The production of a new drug to go into the clinical trials and to sell it is becoming increasingly relevant for M.D. companies. There is little room for delays after a launch date has been set. In general, time is the most significant concern, as one interviewee put it. It takes a lot of time for a time out here to weigh. This results in little tolerance to process adjustments that could lead to threats to compliance and thereby impact times in the market for the software development process. Despite a particular initiative for process improvement to automate a production line, which led to time-to-market delays.

Throughout the new product launch cycle, concentrations of time on the market have often generated organizational resistance to shift. In an M.D. environment,[27] this may be typical: "Any big shift in culture or process can be challenging in a world like this. This is also difficult because of the inertia of a big organization. During an interview, it was suggested that they could increase their performance if they spent more time looking for newer technology and innovations. This is very rare that new equipment is built and tested. It involves the SDLC, as they work with the implementation community, as opposed to the "Tele waterfall form of procedure," An iterative approach would tend to be more successful for them. But they find ways to reduce the overall cycle and to bend the laws of the cycle along the way.

## **VII. DISCUSSION**

Our work revealed some interesting effects on software creation in an M.D. facility, which involves some components, such as regulations. The company's introduction of a procedure may be a potential consequence of compliance with regulatory requirements. From an M.D. company's point of view [27], we firmly believe that a reliable process is an essential element in maintaining high quality in our world of security-criticality. The side effect of this strong conviction may be that it is too reliant on prescriptive, prescribed rules for the process, which require a team approach to discipline. While a well-defined method would be helpful, shooting the target seems easy. A process that isn't sufficiently scalable to cover every part of the company, for example, the research and development portion, requiring an iterative kind of process, leads to dissatisfied workers and maybe a short process break.

### **Leaning the Software Development Process**

It is critical that an organization's work ethic is constantly improved. Many models for process improvements, including CMMI Level 5[28] and ISO / IEC 15504-5[29], are referred to as core elements in the highest degree of process maturity. The lean theory of achieving excellence is close [13]. This ethic is evident in MedTech, where various method changes have been constructive to solve problems. For example, when quotes of potential projects are required to alter activities, a mechanism implemented by MedTech that helps to minimize this is that an internally based resource is available. The method asks a number of critical questions and then offers a high estimate of the cost of the project. The number maybe 20 or 30 percent out, but this is simply to give a magnitude order to people. It allows managers to determine whether the project is to be followed more closely and, therefore, to reduce the amount of value-added project assessments in which developers participate.

The software developers may also follow the same approach. One of the theses proposed by [30] is the fact that developers are inspired to adapt and continue from a continue with realistic projects based on their familiarity with software enhancement. The other activities are defined by MedTech developers and can contribute to process productivity, such as enhanced code reuse, code analysis, testing methods, and development of skills. [30] Some of these activities are independent of their operating regulatory climate, but some form of organizational support is required to find the time for employees to take these initiatives in the lean world of development [13]. The aim is to

find consensus on some specific transitional milestones [and] to define the people and resources to complete the projects. Some changes in software development may be problematic in an M.D. business, such as a source code refactor [31] can lead to previous (and likely certified) code [32] unexpected results [33]. There must be a certain degree of vigilance.

MedTech incorporates a strong process management job ethic, and that isn't any different in the software development community. There's an understanding that people are trying to do it differently. It's very annoying that anyone who uses this tool spends more time recording and reviewing stuff. They constantly evaluate their current method, making [it] more powerful, productive, and leaner than what we are currently doing. It is easier to say than to do, especially in the SDLC.

Lean, which has its roots in a manufacturing environment, has been popular only slightly in areas like software development. Nonetheless, we can see that MedTech has instigated activities that have made some incremental changes to the method using the principle of lean software creation. The knowledge that is hidden by modularization and componentization, which reduces complexity, parametric that render modules less sensitive to implementation and code reuse, are suggested as lean practices [17]. It seems MedTech has started building a library of digital items and reusable parts in the right direction. As one interviewer reported, "I personally see it much better from the testing side.

In terms of coordination between the software community or the more equipment-centered community, they have minimized waste. It was noticed that the two groups fused under the same boss after a recent reorganization and that several more interactions took place. The developers themselves have also begun to cross-train each other on their own initiative so as to increase the level of experience within the community. They now swap roles so that sometimes the guys do machines and sometimes or who do databases, displays ... something of this sort.

For MedTech, more options, such as a TDD approach, are worth exploring [20]. TDD, in combination with their hardware dependencies, may allow major changes in the processes by incorporating test techniques that decouple hardware such as stuffing, mocking, and stubs [38], and hardware simulators [39].

## VIII. CONCLUSION AND FUTURE WORK

Having looked at SDLC in the M.D. production plant, we have established several major factors of impact, namely regulation and concentration of industry, and we have focused on two in this paper. Whilst many M.D. companies are worried about regulatory enforcement, in this case, we saw a feeling about how it can lead to a noisy process in all aspects of the life cycle of the product. Four of the main drivers (Fig. 3) were also listed, and two of the drivers increased again: cost and time to market. The MD industry seems to be getting much more competitive and more cost-driven, and businesses, therefore, are trying to develop their processes, while ensuring compliance with regulations. Since the method for evaluating how much procedure is sufficient does not seem to be available, a detailed review of current systems would be helpful, which may suggest the implementation of too much rigor and thus the likelihood of reducing the amount of work needed.

Lean, while not yet very well established, software development offers the ability to transfer lean manufacturing into the production of software and thereby to gain some of the advantages of lean strategies in other areas of the enterprise. This is useful is a standard of reference in the lean framework that can be used without impacting compliance with regulations.

## REFERENCES

- [1]. E.U., Council Directive 93/42/EEC of the European Parliament and of the Council, Concerning Medical Devices, E. Council, Editor 1993, Official Journal of the European Union.
- [2]. E.U., Directive 2007/47/EC of the European Parliament and of the Council, 2007, Official Journal of the European Union.
- [3]. FDA. U.S. Food and Drugs Administration. Available from: ([www.fda.gov](http://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](http://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/](http://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](http://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.