Regulated Software Development – An Onerous Transformation

Oisín Cawley[1*], Xiaofeng Wang[2], Ita Richardson[1]

[1] Lero-The Irish Software Engineering Research Centre,
University of Limerick, Ireland.
{Oisin.Cawley, Ita.Richardson}@lero.ie

[2] Free University of Bozen
Bolzano, Italy.
xiaofeng.wang@unibz.it.

Abstract. Software development within regulated settings is becoming more and more common place. Compliance typically involves saying what you do and doing what you say. However, in some domains, especially safety-critical ones, it needs to be more than simply following the rules, and should be something which everybody in the organisation supports in their daily tasks. This can be difficult to achieve and requires an organisational transformation, but once begun, sets the foundation on which the software development process can evolve.

Keywords: Regulated Industry Standards, Software Process, Software Quality, Medical Information Systems

1. Introduction

Many years ago there was an advertisement on Television which went something like, "If you had the only car in the world, you could drive where you like, but you haven't so you can't". It was obviously a road safety commercial to re-enforce acceptable rules of driving, but this is very similar to what we are experiencing in the world of regulated software development today. If your application was the only software running, with no interaction with other systems, and no possible effect on third parties then who cares how you develop it?

However, this is not the case for an increasing number of companies. Industries such as the financial services, safety-critical domains like aviation and medical devices, and companies listed on the U.S. stock exchange have seen the conditions under which they

* Corresponding author. Oisin.cawley@lero.ie. Tel: +353-87-4198075

operate change significantly in recent times. These changes have affected the software development processes within such companies by introducing software compliance rules. These additional requirements may take various forms depending on the type of regulations which apply. What is important to know is that they introduce some significant changes which not only can be time consuming and expensive for the software development teams, but have a wide affect within the broader organisation.

Our research has shown that the process changes which must be introduced to satisfy the regulations, transform peoples' daily activities and therefore need to be given the attention and care due of any transformation process. The focus or intention of the regulations is important. A clear understanding of this will drive: the design of your new development processes, where a concentrated change management process will be needed, and the type and level of detailed evidence to be maintained for those crucial and sometimes painful audits.

2. Research Process

In this paper we have drawn on research into regulated software development from both a financial and safety-critical perspective.
One author has 11 years experience working in a large US multinational (MyOrg) [1], and 7 of those under the Sarbanes-Oxley (SOX) regulations (www.soxlaw.com). MyOrg is a leader in global supply chain business process management. NASDAQ listed, it has over 25 centres in 14 countries and has a diverse and interconnected collection of information systems which are developed and maintained by a combination of in-house and outsourced personnel. By means of reflective analysis we examined the effects on the software development and support teams when the SOX regulations came into effect in 2002.
With a growing interest in modern software development methodologies (SDMs) such as Agile and Lean, we performed a mapping study of these SDMs within safety-critical regulated domains [2]. The resulting state-of-the-art gave us a solid understanding of the development issues in such domains as Aviation, Automotive, and Medical Devices.
The Medical Device industry is seen as a growth area in many countries, and is fast developing into a race for world leadership. As an industry, which was somewhat immune to cost pressures, it is now feeling the effect of the demand for lower cost products. By means of an in-depth case study at a medical device manufacturing plant (MedTech), we examined the issues affecting the software development and which typically lead to the adoption of heavy SDMs [3]. MedTech is a US multinational with over 25,000 employees and develops medical solutions. A series of semi-structured interviews brought to light the concerns people had with the software development life-cycle (SDLC) and how the regulations had impacted it.
We have synthesised this data into a model of influences which is presented in this paper.

3. The Effects of Compliance

The SDLC within a regulated area is reflective of a number of key influences. We synthesised our findings into a common model, categorising the influences into 4 groups. These categories together with some of their main influences are depicted in Figure 1, with a selection of exemplars included for each.

3.1 Regulations

At the top of Figure 1 we have the first of these contextual elements, regulations. Why do we have regulations and how important are they in terms of the development of medical device software? According to [4], regulations are simply a form of social organisation, and therefore supports the definition of regulations used in this study as being: "*rules, principles, or conditions that govern procedure or behaviour*" [5]. There are those who argue that there is too much regulation and that not enough research has been done in assessing the adequacy of regulations in achieving their intended aims [4]. For example, regulations governing financial institutions, such as the Basel Accord [6] and the Sarbanes-Oxley Act [7], did not prevent the global banking crises of 2007.

There are, however, different types or levels of regulation ranging from low (self regulation), medium (Government regulation), to high (Litigation) [8]. It is only when a lower form of regulation is seen to be ineffective is it raised to a higher level. Nonetheless, it is difficult to argue against high levels of regulation which aim to ensure human safety. For example, the standards and recommended practices issued by the Association for the Advancement of Medical Instrumentation (AAMI), aspire to a "*continued increase in the safe and effective application of current technologies to patient care*" [9], a laudable objective. While there is little debate about the merit of such motives, the AAMI does however have an additional objective, namely "*the encouragement of new technologies*" (Ibid). There is therefore a balance which needs to be struck between this drive for innovation and need for a high degree of device safety and the consequent higher level of regulation [10].

Each country has its own specific regulatory requirements when it comes to medical device software. Within the United States it is governed by the Code of Federal Regulations Title 21. Within the European Union (E.U.) it is the Medical Devices Directive. More and more, such federal documents allow for adherence to international standards such as [9, 11-12], as satisfying these requirements. It is the influences, affected by such international standards and guidelines, which make the medical device software development process unique. Our research has found that medical device companies predominantly employ, what can be described as, heavy weight software development methodologies to ensure all required steps are taken to satisfy regulatory requirements.

Within our model, the arrows emanating from the Regulations box are shown leading to the other three categories, indicating that compliance with the regulations must be addressed within multiple levels and contexts.

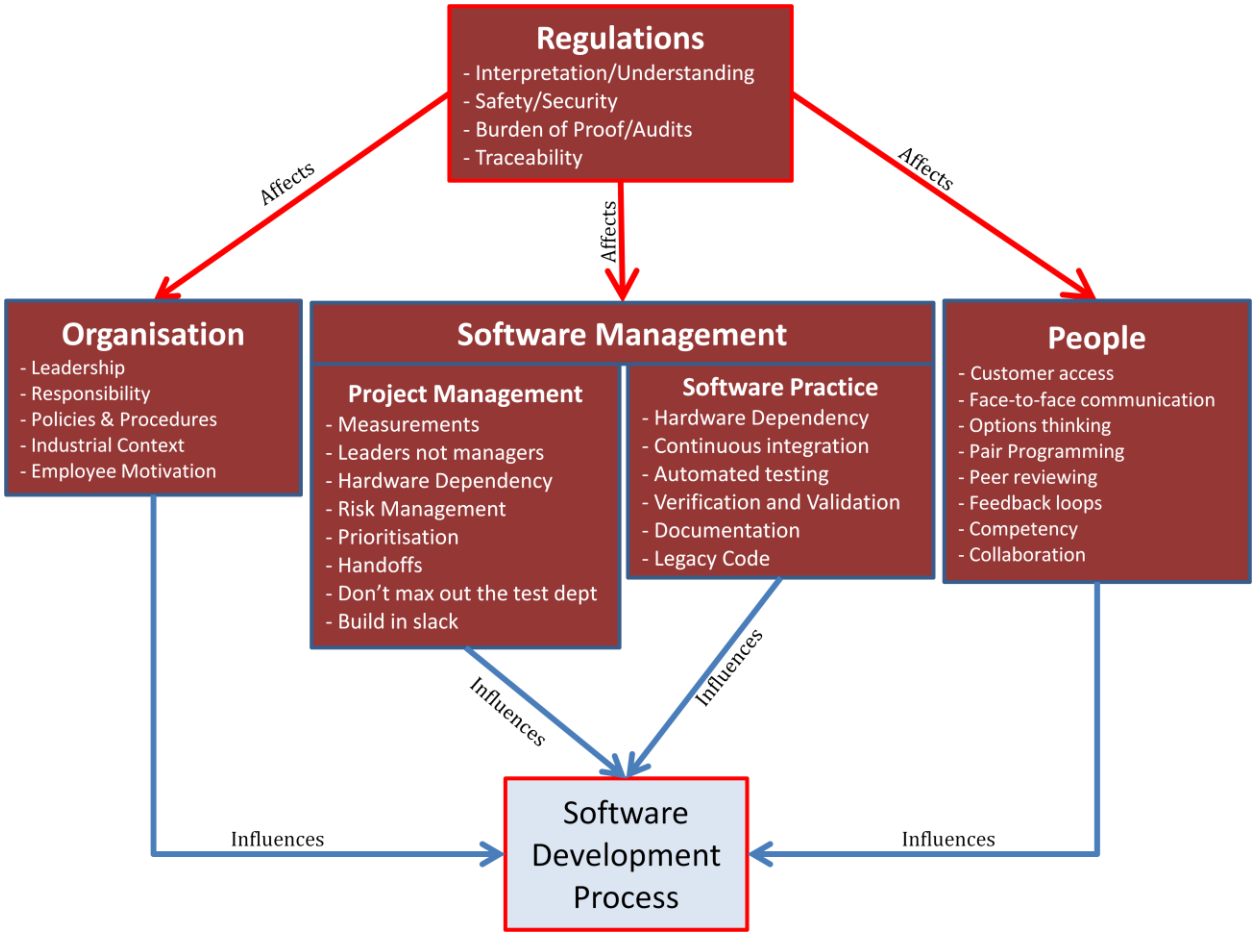

Fig. 1. Categories of Influences on the SDLC within a Regulated Context

3.2 Organisation

The term 'Organisation' can be a nebulous term and so for clarification the following definition is used: "*[An] organisation is a complex social system and is the sum of many interrelated variables. The operations of the organisation are influenced by the external environment of which it is part*" [13].

The relevance within our theoretical model is that the organisation influences the software development process by defining development tasks and delegating roles (such as developers and testers), responsibilities (such as project managers and validation engineers), and authority (such as approvals). In addition, its relationship with the surrounding environment is an important aspect relevant to the model as identified by the link with the regulations box. The regulatory context will influence the organisation's ethos in terms of ensuring safe and secure software, the workforce's attitude to risk management, and their sense of responsibility.

For example, within the MD regulations, the international standard ISO 13485:2003 [11] calls for a documented quality management system. In large companies, that will be satisfied at multiple levels. Firstly, the organisation may develop and publish corporate policies and procedures which describe the various processes at a high level. This is then complemented by specific standard operating procedures (SOPs) at the business unit level, and/or a software development process documents detailing software practices. Lower level processes which align with higher level ones demonstrate a coherent and unified approach, across the whole organisation, to addressing the regulatory requirements.

The organisational guidelines and supports for the software development process often require significant investment. To protect that investment, an organisation will often seek to employ a formal software development methodology which is future proof (teachable), provides consistency, generates explicit deliverables, and provides an engineering-like development discipline [14]. The organisational influence plays a crucial role when it comes to the implementation of such a software development process or implementing changes to an existing process, for example, when entering a regulated industry. Such process changes may affect the way people do their jobs or indeed the job descriptions themselves, and since changing work practices is not a trivial undertaking, it should be dealt with like any organisational transformation [15], [16].

3.3 Software Management

The software management box incorporates the overlapping activities of managing the tasks, resources and schedules, combined with the specific practices (many of which are of a technical nature) needed to perform the various activities within the SDLC. This category naturally influences the SDLC since it will be within these competencies, capabilities and situational contexts that the SDLC will need to be framed. For example, the technical nature of the product will automatically dictate the type of skills required and the type of development environment needed. The availability or lack or availability of these resources will shape the resulting SDLC [17]. In addition, the existing technical infrastructure will go some way towards assisting or hindering the adoption of a specific

SDLC approach. For example a company which uses a language not conducive to an object oriented approach [18] (such as earlier versions of Visual basic, Fortran and Pascal), may have difficulty in following an SDLC which calls for such practices. Similarly, unless some investment is made in additional hardware and/or software, an SDLC which promotes test driven development [19] and continuous integration [20] is unlikely to be proposed where the environment is not adequately equipped.
Different people and organisations approach project management of software development differently. Depending on the perceived importance of the software, for example is it seen as a strategic competitive advantage [21], [22], or merely an asset to be managed [23], [24], the SDLC will reflect this. A company which sees the software as being strategically important may also be more supportive of pursuing lean or agile approaches such as iterative development [25] and/or closer contact between developers and end users [26] in order to improve that key process area.

3.4 People

An obvious influence on the software development process are the people who are involved with it on a daily basis. There are a myriad of areas which have been studied over the years surrounding the issues with software development due to the human condition. For example the fundamental problems associated with knowledge management in such a specialist environment continues right up to the present day [27], [28], [29], [30]. Another factor is the motivation of the software developers themselves [31], [32], [33], something considered to be the most impactful on productivity [34]. Indeed unmotivated developers can be seen as sources of negative productivity and a liability to a project's success [35].
The effect of regulatory compliance is very notable at a personnel level as it is precisely the human activities that are being governed. When moving from an unregulated into a regulated environment, unless the work processes are already fulfilling the regulations (experience suggests that this is unlikely), there is a need for peoples' daily activities to change. For example, both SOX and MD regulations look for some level of independence in certain key areas. SOX looks for segregation of duties when it comes to code deployment or even access to a production system, while MD regulations expect independence between developers and validation engineers. The typical approaches to activities such as communication and knowledge transfer, where important ad hoc conversations go undocumented, or an approval is given verbally, are no longer acceptable. When people are used to operating in an environment where issues can be fixed "on the spot", these tighter controls can be very frustrating for both the technical employee as well as for the person awaiting resolution.

4. A Transformation Process

In 2007 the European Medical Device Directive (MDD) expanded its definition of a MD to include stand alone software in its own right to be a possible MD [36]. As was the case when SOX was enacted in 2002, this is where a transformation process will be necessary.

In other words, existing processes and work habits need to change, and as with any change process requires disciplined attention to some important aspects [16].
This transformation is not only a change in physical activities such as recording test results or maintaining traceability, but equally requires a change in mindset. For example, the MD regulations hold risk management (RM) (safety risk as opposed to, for example, project risk) as a critical aspect of the product development:

> "*The manufacturer shall establish, document and maintain throughout the life-cycle an ongoing process for identifying hazards associated with a medical device, estimating and evaluating the associated risks, controlling these risks, and monitoring the effectiveness of the controls*" [11].

The words "throughout the life-cycle an ongoing process" means it is not something that starts and finishes during a particular phase of the life-cycle but is something that must be woven into the entire process. From a software point of view, RM is completely irrelevant without the context of the surrounding device or people and processes. A software failure alone cannot cause harm. It is important therefore that the interface between the software team and other teams, such as the hardware developers and quality engineers, supports an ethos of thinking about hazards in a cross functional holistic sense at all times (Figure 2).

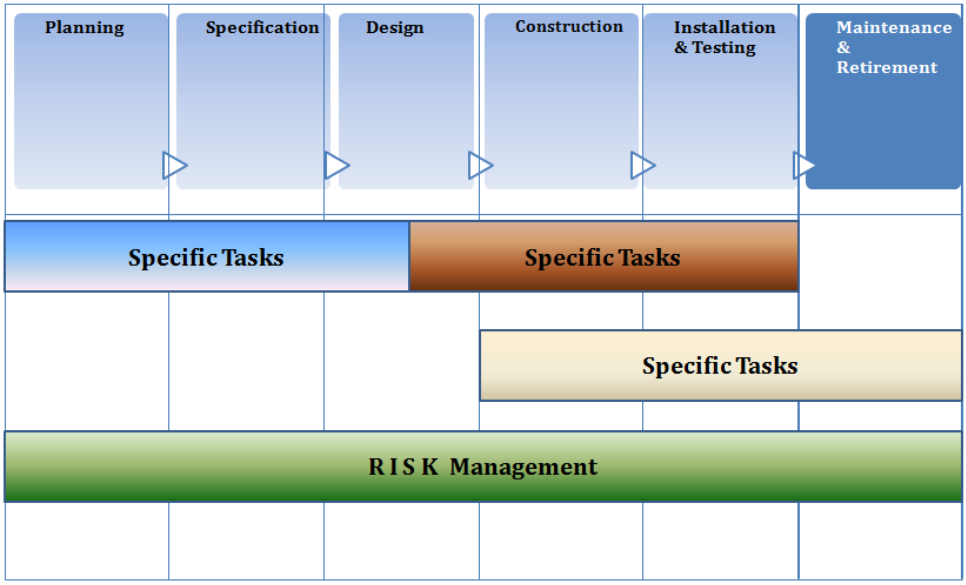

Fig. 2. SDLC with RM across the entire process

In order to structure the discussion around such a transformation we utilise the four categories identified in Figure 1 and draw on specific examples for clarity.

4.1 Organisation

Changing work practices, in order to be compliant, requires the generation of a sense of urgency typical of a transformation process [37]. However, if this urgency is not communicated in a timely manner, it can lead to severe teething problems when it comes to being audited. Within MyOrg, the first dry-run SOX audit presented some non-conformances because there was no evidence of compliance dating back to the official launch date. Although evidence could be collected from dates after this, the issue resulted in a painful retrospective evidence gathering exercise.

Similarly, within the MD regulations, ISO 13485:2003 [11] states that management has the responsibility of "*communicating to the organisation the importance of meeting customer as well as statutory regulatory requirements*". MedTech's approach to this was to develop a mandatory training schedule for all employees on the newly formed product development process, administered by the human resources department.

Leadership is an important attribute in a transformation process. Especially within larger organisations, the need for a strong guiding hand is required [37]. This need manifests itself at a number of levels, however, certain important roles and responsibilities will be mandated by the regulations. SOX, for example, holds the senior executives as individually responsible for the accuracy of the financial reports. MD standard ISO 13485:2003 holds "*Top Management*" responsible for "*evidence of its commitment to the development and implementation of the quality management system*" and for the implementation of an ongoing risk management process. These typically translate into the formation of specific roles within the company and the delegation of authority to ensure their implementation. MyOrg instituted a global SOX compliance officer, while MedTech formed a quality assurance department which took guidance from a corporate compliance group responsible for ensuring internal processes were compliant.

A final comment from an organisational perspective is to realise the need to evolve the policies and procedures once the initial transformation is complete. No matter how well designed and executed this is, it will be received with mixed opinions. It is incumbent on senior management to be aware of and solicit this feedback in order to improve. Within MedTech, they carried out "*an upgrade to the corporate process because we've been using it for almost 3 years at this stage now. So we've gotten feedback from the different sites and we've released a common overarching software policy that governs the whole corporation*" (a senior software quality engineer). Similarly, within MyOrg, they reviewed the processes internally one year after SOX was introduced, leading to a revision of the process documents, thus reducing the associated overheads.

4.1.1 The Bottom Line

If the focus of a organisation is to make money [38], then the cost of operations is a critical concern. Compliance costs money. These costs come in many forms, such as employee training, longer project timelines, additional verification & validation (V&V), audits (internal and external), data archiving and staying on top of regulatory updates.

Within MyOrg, the software development group automatically added 20% to project estimates to account for the extra activities. This reduced as time went on and processes got more embedded in daily activities but from a business management point of view caused some alarm as project costs jumped. MedTech had a similar experience which led to many projects not proceeding at all.

The requirements around V&V, for example within the MD regulations, in addition to adding cost, also cause confusion. The confusion resides with the regulatory documents, in that they are necessarily broad based and unspecific in terms of how V&V should be performed [39]. In fact this is a common criticism of the regulatory documents in general: "*I think the regulations haven't reached full maturity in terms of what's needed. Certainly the FDA regulations, the actual text of it, is a couple of paragraphs. A lot of it is interpretation after that*" (Senior Engineer, MedTech). Because of this lack of clarity and the risk averse nature of the business units, there is a tendency to do too much in order to be compliant. Speaking about their validation process, a MedTech project leader said "*Personally I think we over club it a lot of the time*". Not only can the physical validation be overdone, but the documentary evidence required around it generates a lot of concern: "*We're caught in this deadlock of paperwork... The physical work is costing me X, and 3 to 4X is what it's costing me to actually fill out the paper work to validate it*".

While cost pressures continue to increase, this additional cost is an unwelcome burden. However, this reality needs to be reflected in the expectations of the business management teams. It also behoves the software development teams to find ways to streamline these new tasks to minimise cost. The following exemplifies the point: "*Since [the process introduction] we've been looking and going, 'Oh my God', this has nearly crippled the business, we need to Lean it and take it all back out again*" (Principle Engineer, MedTech). Software development techniques, such as test driven development, automated testing, continuous integration, and automatic document generation can aid in simplifying the processes. They can support finding and eliminating defects as early as possible, and can also help ease the transformation process by implementing a more lean and agile approach [3].

4.2 Software Management

Depending on the industry, the regulations will have different levels of tolerance to compliance. Moving from unregulated to regulated requires an understanding of that level of tolerance. Within the MD domain, for example, there is little tolerance when it comes to demonstrating compliance "*If you're Microsoft you can release something that has a tolerable quality level but it still has problems with it. Whereas in our environment it's a lot more, you cannot go any further until you have done this. You must tick 100% of the boxes not 95% or 80%*" (MedTech Senior Engineer). Within safety-critical companies, management therefore have little appetite when it comes to changing established processes for fear of falling foul of the regulators. Nevertheless, the modern business environment is calling for more efficient and cost effective processes, and so change is inevitable.

To ensure compliance, a common approach to software project management is a phase-gate process. The salient point is the identification of specific development phases with clearly defined entry and exit criteria. Only when these criteria have been fulfilled can you enter/progress to the next stage. This approach seems to be a consequence of the regulations "*The regulations have driven us, or the interpretation of the regulations have driven us towards meeting particular gates or milestones or particular steps*" (MedTech Senior Engineer).
The regulations, however, do not prescribe any particular methodology, even though they may look and read like they favour such a waterfall-type approach. For example, ISO14971:2007 [12] states that "This standard does not require a particular software development life-cycle". What this allows for then, is the potential to evolve the development process and look at more efficient approaches. Our research has seen that companies are looking to the advantages that might be gained from agile methodologies such as eXtreme Programming (XP) and Scrum and, consequently, finding the evolution of their SDLC less problematic than anticipated [2].

Evolving your company's software development process, typical in un-regulated industries, should be no different within regulated domains. In fact, we would suggest that the influences of the regulations lead to the adoption of inefficient development processes. This can be attributed to a lack of clarity/understanding and fear of non-compliance. Once the initial process transformation occurs and becomes embedded, we suggest that companies will start finding weaknesses and will need to adjust their software processes accordingly. These changes will be in many forms such as more effective resource scheduling or implementing lean and agile practices. Within MedTech, they reduced the time for a typical validation process by 30%: "*When we started out, there were validations that were effectively ongoing for 6, 9 months that just got bogged down and ran into problems ... now we were saying we're going to do this and we're going to complete it in 3 months*". By realising that resources were being inefficiently scheduled, they examined ways to address it, such as load levelling. Similar to the Kanban system in lean manufacturing [40] and using cards to represent different activities within the software project, they identified where bottlenecks were occurring: "*The card represents something, so in my case it was representing a validation protocol or validation execution*" (MedTech, Senior Engineer). By making the issues visible it was easier to discuss and address them "*It was a bit like clearing the jam in the pipe. Once you got them moving you got a big flow, it made a big difference*".
In addition, being a little less dogmatic when it came to fully completing documentation, allowed them to achieve faster prototype systems without breaching regulations: "*... we discovered through pain that we need to give them a piece of equipment and let them run it for a while and they come back and tell you what they actually really want*".

4.3 People

An organisation can't change unless its people change [41]. One way to achieve this is to issue new SOPs for project activities. Before rolling these out, however, it is important for

employees to be brought up to speed on why there is a need for change and how the new SOPs satisfy that need. If this is not clearly enunciated then people may look for shortcuts. As a software developer at MedTech put it: "*Before, we thought the system wasn't great and this [the new process] came along which probably was worse. I don't know how this came about*". This lack of clarity/buy in can result in short cutting or working outside the defined process: "*You find that there's an awful lot of background work done before we start the development proper...We officially don't know about that*" (Senior Software Quality Engineer, MedTech).

A further difficulty with transforming work practices is coping with legacy systems. Older products, developed using previous processes and practices, may not be suitably designed or have available the requisite artefacts to suit the new process. In this case, software modifications may have to be performed differently depending on the project. In addition, if an employee is assigned only to such legacy systems, then, despite having been fully trained, they get very little exposure to the new processes and are potential liabilities if assigned to projects using the new approach. As stated by a MedTech software developer about the new process: "*It's 2 years here in [MedTech] that they've started doing it. I actually haven't worked on a project yet, in the development phase that has used it*".

Transforming employee work practices is therefore a protracted affair and requires careful implementation and monitoring until the practices start to get embedded: "*Once you get to know what you need, and the order of things, after that period of time it becomes second nature*" (MedTech Software Developer).

5. Discussion and Recommendations

Regulatory compliance from a software development perspective can be daunting for a company which is unfamiliar with it. The effects of regulation are widespread within an organisation and, due to the inexact nature of the regulatory documents, compliance can lead to inefficient and heavy development lifecycles.

This of course does not have to be the case. It is important to really understand the intention of the regulations, and once that is crystal clear, define the processes accordingly. We should remember that the people best suited to determining how to achieve the objectives of the regulations are the people who work in these specific contexts.

Each company is different and so a one-size-fits-all approach will not work. Rolling out an umbrella policy or procedure across an entire organisation will lead to inefficient work practices. Even within companies, different departments will need to have flexibility in how they shape their processes. For example, the research and development department needs to be free to innovate without getting bogged down in red tape. In addition, because the software development process is a collaborative process, the overlapping between departments needs to be smooth.

We should remember that many of the activities mandated by the regulations are probably already being performed to some degree. There is nothing ground breaking in what they look for, just a more robust method of ensuring the right things are happening. Becoming compliant therefore requires a careful transformation process which takes a multi-layered view. Once this initial transformation effort happens, the way is paved for an improvement process which evolves the SDLC.

We conclude with a series of recommended steps, shown in Figure 3, which can assist the successful introduction of regulatory compliant software development processes within a medical device organisation.

Fig. 3. Suggested steps for regulated process implementation

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 10/CE/I1855.

References

1. Cawley, O., Richardson, I.: Lessons in Global Software Development – Local to Global Transition within a Regulated Environment. European Systems & Software Process Improvement and Innovation. Springer, Grenoble, France (2010)
2. Cawley, O., Wang, X., Richardson, I.: Lean/Agile Software Development Methodologies in Regulated Environments – State of the Art. In: Conference Lean/Agile Software Development Methodologies in Regulated Environments – State of the Art, pp. 31-36. Springer Berlin Heidelberg, (2010)
3. Cawley, O., Richardson, I., Wang, X.: Medical Device Software Development - A Perspective from a Lean Manufacturing Plant. In: O'Connor, R., Rout, T., McCaffery, F., Dorling, A. (eds.) 11th International Conference on Software Process Improvement and Capability Determination, pp. 84-96. Springer, Dublin, Ireland (2011)
4. Campbell, M.K.: Regulations. IEEE Potentials 23, 14-15 (2004)
5. The-Free-Dictionary: Regulations. (2011)
6. http://en.wikipedia.org/wiki/Basel_II_Accord. Last Accessed 31-July-2012
7. http://www.sec.gov/about/laws.shtml#sox2002. Last Accessed 31-July-2012
8. Bush, W.R.: Software, regulation, and domain specificity. Information and Software Technology 49, 44-54 (2007)
9. ANSI/AAMI/IEC: 62304:2006 Medical Device Software-Software life cycle processes. pp. 67. Association for the Advancement of Medical Instrumentation (2006)
10. Ziegler, A.S.: Regulation. Annals of the New York Academy of Sciences 1093, 339-349 (2006)
11. ISO: ISO 13485:2003 Medical devices -- Quality management systems -- Requirements for regulatory purposes International Organisation for Standardisation (2003)
12. ISO: ISO 14971:2007 Medical devices -- Application of risk management to medical devices. International Organisation for Standardisation (2007)
13. Mullins, L.: Management and Organisational Behaviour Prentice Hall (2004)
14. Roberts, T.L., Jr., Gibson, M.L., Fields, K.T., Rainer, R.K., Jr.: Factors that impact implementing a system development methodology. Software Engineering, IEEE Transactions on 24, 640-649 (1998)
15. Kotter, J.P., Schlesinger, L.A.: Choosing strategies for change. Harvard Business Review 57, 106-114 (1979)

16. Small, A.W., Downey, E.A.: Managing change: some important aspects. In: Conference Managing change: some important aspects, pp. 50-57. (2001)
17. Kettunen, P., Laanti, M.: How to steer an embedded software project: tactics for selecting the software process model. Information and Software Technology 47, 587-608 (2005)
18. Poppendieck, M., Poppendieck, T.: Lean Software Development: An Agile Toolkit Addison-Wesley Professional (2003)
19. Beck, K.: Test Driven Development: By Example Addison-Wesley Professional (2002)
20. Hibbs, C., Jewett, S.C., Sullivan, M.: The Art of Lean Software Development. O'Reilly Media (2009)
21. Porter, M.E.: Competitive advantage: creating and sustaining superior performance : with a new introduction. Free Press (1998)
22. Prahalad, C.K., Hamel, G.: The Core Competence of the Corporation. Harvard Business Review 68, 79-91 (1990)
23. Ben-Menachem, M.: Towards management of software as assets: A literature review with additional sources. Information and Software Technology 50, 241-258 (2008)
24. R, T.: Untangling intangible assets. OECD Observer 13-15 (2011)
25. Rasmussen, R., Hughes, T., Jenks, J.R., Skach, J.: Adopting Agile in an FDA Regulated Environment. Agile Conference (2009)
26. Rottier, P.A., Rodrigues, V.: Agile Development in a Medical Device Company. Agile, 2008. AGILE '08. Conference, pp. 218-223 (2008)
27. Robillard, P.N.: The role of knowledge in software development. Commun. ACM 42, 87-92 (1999)
28. Damian, D.E., Zowghi, D.: The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization. In: Conference The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization, pp. 319-328. (2002)
29. Ye, Y., Nakakoji, K., Yamamoto, Y.: The economy of collective attention for situated knowledge collaboration in software development. Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering, pp. 109-112. ACM, Leipzig, Germany (2008)
30. Levy, M., Hazzan, O.: Knowledge management in practice: The case of agile software development. In: Conference Knowledge management in practice: The case of agile software development, pp. 60-65. IEEE Computer Society, (2009)
31. Burn, J.M., Couger, D., Ma, L., Tompkins, H.: Motivating IT professionals-the Hong Kong challenge. In: Conference Motivating IT professionals-the Hong Kong challenge, pp. 524-529 vol.524. (1991)
32. Sharp, H., Hall, T.: An initial investigation of software practitioners' motivation. Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering, pp. 84-91. IEEE Computer Society (2009)
33. Treude, C., Barzilay, O., Storey, M.-A.: How do programmers ask and answer questions on the web? (NIER track). Proceedings of the 33rd International

Conference on Software Engineering, pp. 804-807. ACM, Waikiki, Honolulu, HI, USA (2011)
34. Boehm, B.: Software Engineering Economics Prentice Hall (1981)
35. McConnell, S.: Problem programmers. Software, IEEE 15, 128, 127, 126 (1998)
36. McHugh, M., McCaffery, F., Casey, V.: Standalone Software as an Active Medical Device. In: O'Connor, R., Rout, T., McCaffery, F., Dorling, A. (eds.) 11th International Conference on Software Process Improvement and Capability Determination. Springer, Dublin, Ireland (2011)
37. Kotter, J.: Leading change: Why Transformation Efforts Fail. Harvard Business Review 73, (1995)
38. Goldratt, E.M.: The Goal: A Process of Ongoing Improvement North River Press (1992)
39. Vogel, D.A.: Medical Device Software Verification, Validation and Compliance Artech House Publishers (2010)
40. Anderson, D.J.: Kanban. Blue Hole Press (2010)
41. Mathiassen, L., Ngwenyama, O.K., Aaen, I.: Managing change in software process improvement. Software, IEEE 22, 84-91 (2005)

Biographies

Oisín Cawley is a doctoral researcher with Lero in the University of Limerick. His research interests are in software processes within regulated industries, especially the Medical Device domain, Lean\Agile methodologies, and Global Software Development. He has worked for 17 years in software development within industry and holds an MBA from Dublin City University, Ireland.

Dr.Ita Richardson is Senior Lecturer at the University of Limerick, Principal Investigator within Lero and Competence Leader for Lero's Software Process Improvement and Global Software Development research. She researches software processes in medical device, automotive and financial services domains, and within global software development environments and SMEs.

Dr. Xiaofeng Wang is a researcher at Free University of Bozen-Bolzano. Before that she worked as a post-doctoral researcher in Lero, the Irish software engineering research centre. Her research areas include software development process, methods, agile software development, and complex adaptive systems theory. Her doctoral study investigated the application of complex adaptive systems theory in the research of agile software development.