Regulatory Compliance and its Impact on Software Development

Abdelwahab Hamou-Lhadj
Software Compliance Research Group
Department of Electrical and Computer Engineering
Concordia University
1455 de Maisonneuve Blvd. West
Montréal, QC, Canada
abdelw@ece.concordia.ca

Abstract

Abstract. Since the outset of our research, we have been looking for ways to help software companies develop software systems for which regulatory compliance is a key quality attribute. This is because many regulations, laws, standards, and other authoritative rules have a direct impact on the way software systems, used by regulated organizations, are developed, tested, and maintained. In this paper, we particularly discuss the impact of regulatory compliance on software development practices. We achieve this by discussing how regulations and laws impact three main aspects of software which are: product, process, and project. We argue that there is a need for a new field of study that we refer to as software compliance engineering and which aims to investigate techniques and tools for the development and maintenance of auditable software systems.

1. Introduction

Consider the following scenario:

A North-American public software company that delivers IT solutions to health institutions must comply with the North-American laws that protect patient private information such as the Health Insurance Portability and Accountability Act (HIPAA) if the company operates in the U.S., or its Canadian equivalent, the Personal Health Information Act (PHIA). It also needs to comply with various security laws and standards to protect sensitive information such as patient personal records. Examples include the ISO 27000 standards, the Privacy, Cryptography and Security Gridlines issued by OECD, the Privacy Act in the U.S., or the Personal Information Protection and Electronic Documents Act (PIPEDA), in Canada. If the company uses (or develops) software systems to control medical devices then it should also adhere to the (stringent) requirements imposed by the Food and Drug Administration (FDA) regulations on the process by which medical device software is developed. Being a public trade company, U.S. laws require that this company comply with the Sarbanes Oxley Act (SOX) and perhaps the Gramm-Leach-Bliley Act (GLBA), which govern the proper reporting of financial statements. If the company has a client base that spreads over many countries, then it should not only comply with the laws and regulations of these countries but also consider the many agreements that regulate international trade such as NAFTA. And, the list goes on.

Complying with the law and regulation is often a tedious task, especially for global software companies with a client base from many industry sectors, spread over different geographical areas. First, the number of regulatory compliance requirements that need to be considered can be considerably high as reflected, though at a very small scale, in the above scenario. In his description of the U.S. laws that regulate data privacy and security alone, Silverman notes that "Given the enormous breadth of federal regulation, it is not possible to catalogue the full range of U.S. law pertaining to data, privacy, and records" [1]. In addition, laws are often developed by different legislative bodies with little effort for consistency or convergence with similar legal requirements, which often results in duplicated and conflicting rules [2]. Inconsistencies also appear due to the fact that regulations are rarely created from scratch. They often depend on other existing laws, which altogether form a network of provisions in which a modification in one place can propagate to many other places.

Clearly, there is a need to investigate techniques to manage a large number of possibly overlapping or conflicting regulatory compliance requirements [5]. This is the objective of our research, with a particular focus on helping global software companies cope with the increasing demand from regulated businesses for software solutions that satisfy regulations, laws and standards that apply to their industry sector. This is because many regulations and laws have a direct impact on the way software systems, used by regulated organizations, are developed, tested, and maintained. We refer to this field of study as *Software Compliance Engineering, which we define "as the software engineering field that aims to define, develop, and evaluate tools and techniques intended to support the software community in the production of software systems for which regulatory compliance is a built-in quality attribute."*[1] The aim behind this definition is to position "compliance" at the same level as any other software quality attribute such as performance, security, dependability, to name a few, which have been studied extensively in software engineering and led to powerful techniques. We call a software system for which regulatory compliance is an important aspect an *Auditable Software System,* since it requires one way or another (i.e. either internally or by external agencies) to pass an audit process that shows the degree of its adherence to the applicable laws. Hence, we aim through software compliance engineering to understand ways to build, test, and maintain auditable software systems in an efficient manner, using sound techniques (that have yet to be developed).

In this paper, we discuss the regulatory compliance landscape and how it impacts software development. Particularly, we focus on three aspects of software: product, process, and project capabilities. The objective is to motivate and start the discussion on the need for software compliance engineering.

2. Regulatory Compliance and Software Development

The compliance landscape is comprised of various types of documents including regulations, standards, and guidelines [3]. Collectively and throughout this paper, we refer

[1] http://users.encs.concordia.ca/~abdelw/softwarecompliance.html

to these compliance items as *authoritative documents*. Authoritative documents can be created at the local, state, federal, or international level. It is often common to have regulations that overlap in their intents and requirements while many other ones conflict. This appears to be due to a lack of a centralized organization responsible for mapping regulatory compliance requirements. Contradictions might also occur within the same regulatory domain as the law evolves and changes over the years. The new requirements might contradict or otherwise conflict with older ones as noted by D. Vogel when describing the FDA regulations [4]. Vogel adds that "Even when legislation, regulations, and guidelines are not changing, there will always be a spectrum of interpretation and opinion within the FDA" [4].

From the software engineering perspective, one can argue that these are just another set of requirements that the system needs to support. Though this argument has some merit, this is no as evident as it might appear, it should be noted that many regulations (such as FDA regulations) do not impact product functions but rather the process by which the product has been developed. In addition, due to the large number of regulations, there should be a way to prioritize them and ensure that the most important ones are properly handled throughout the entire software project. These three aspects of software development (i.e. product, process, and project) are discussed in more detail in what follows.

2.1. Product

The provisions of many regulations translate inevitably into functional requirements of the end product. For example, a software system that needs to comply with HIPAA requires the presence of several authentication and security mechanisms to protect patient information saved in an electronic format. Similarly, a financial firm will expect that its transactions management system would support the requirements of the Sarbanes-Oxley Act, in particular Section 404, which calls for the creation and maintenance of internal controls over financial reporting. Some of the features that the system needs to support include the proper recording of transactions according to generally accepted accounting principles, transactions are executed in accordance with management authorization, an audit trail, etc.

It should be noted, however, that it is often a serious challenge to translate compliance requirements to software requirements. This is due to the fact that compliance requirements imply, but do not specify, some software product capabilities. In addition, the legal language used to develop authoritative documents is far from the software technical language and terminology: Regulations are written by lawyers, but often expected to be used by managers, auditors, and technical stakeholders such as software engineers with little knowledge of law and law jargon [5].

Many studies that deal with legal requirements and their impact on software development tend to focus on the product capabilities. Perhaps, one of the most comprehensive research projects in this area is the REGNET project [6]. Led by members of the Engineering Informatics Group from Stanford University, the project aims to create an information infrastructure that supports U.S. federal and state regulations. The main outcomes of the project include an XML-based repository to represent specific government regulations, an approach for locating and comparing related regulations based

on information retrieval techniques, feature matching, etc., and a compliance assistance system that facilitates the analysis of the compliance documents in question with the objective of extracting functional requirements.

Another study proposed by Breaux in his Ph.D. dissertation, which consists of the Frame-Based Requirements Analysis Method (FBRAM), aims to extract legal requirements from U.S. regulatory documents. Breaux used domain-independent upper ontology, natural language phrase heuristics, a regulatory document model and a frame-based mark-up language to represent legal requirements, while maintaining tractability between the regulations and the formal representation in the frame based model.

In 2005, OMG[2] (the Object Management Group) created the Governance, Risk Management and Compliance Roundtable (GRC-RT) where experts in the field are invited to discuss ways to help companies deal with the increasing number of regulatory documents and their impact on organizations [7]. The GRC-RT group has initiated many programs that address compliance issues, among which the most related one to this paper is the creation of a Global Rules Information Database (GRC-GRID), an open resource for GRC professionals in which many financial laws are represented and readily available for querying.

2.2. Process

Some compliance requirements do not necessarily impact the product functions, but the process used during the development of the product. Therefore, they are often not considered in the management of product requirements. The FDA regulations[3], for example, impose stringent requirements on the software process by which medical device software systems are developed. The FDA requirements often translate into documenting and following specific guidelines to certify that the system is built, verified, and validated in a systematic manner and according to proven software engineering practices. The FDA requirements impact a large spectrum of software process activities including requirement analysis, design, implementation, maintenance, and testing phases.

Following FDA requirements, a software process should allow the generation of multiple documents (requirement and design specification documents, traceability matrices, test logs, etc.) for each phase of the software process. In addition to this, FDA requires the presence of many software practices such as prototyping, usability analysis, risk management, etc. that many existing software processes do not support. The situation is made worse for agile processes. Mehrfard and Hamou-Lhadj discuss the impact of FDA requirements on agile processes in more detail in [8]. They showed that Extreme Programming (the focus of their study) lacks many activities that FDA requires, which hinders the compliance of software products developed using XP with FDA requirements. They concluded that there is a need to investigate ways to reconcile process agility with auditability.

---

[2] http://www.omg.org

[3] The FDA regulates more than $1 trillion worth of consumer goods, about 25% of consumer expenditures in the U.S. It also tends to issue very several fines for non-compliance.

2.3. Project

Due to the large number of regulations, there should be ways to prioritize them based on their relevance and impact on the final product. This can be done during the project initiation and planning phase. Dealing with different authoritative documents for the same project is however a complex and expensive activity. This is due to the fact that, in order to derive the applicable compliance requirements for a specific software development project, it is required to define the common and shared elements of these authoritative documents. This is done through a technique commonly called compliance mapping. Using this technique requires specific expertise in many different regulatory areas and can be very often time-consuming. The selected regulations should be taken into account throughout the entire project life cycle (i.e. from requirement gathering to release). There should be some key performance indicators (identified at the organizational level) that measure the extent by which regulatory requirements have been implemented. An independent internal assessment activity in the software development life cycle can be added to make sure that regulatory compliance is taken into account as it should be. Other factors that impact proper handling of regulatory compliance are concerned with having an effective training program for software developers. In addition, mechanisms that facilitate the auditing process should also be put in place.

3. Conclusion

The objective of this paper is to open the floor for discussion on the need for a systematic way to deal with regulatory compliance and its impact on software development. We refer to this as software compliance engineering. We discussed how regulatory compliance forms a complex world that can impact in many ways the way software systems used by regulated organizations are developed.

4. References

[1] M. G. Silverman. *Compliance Management for Public, Private, Or Nonprofit Organizations, McGraw-Hill Professional.* McGraw-Hill Professional, 2008.

[2] A. Hamou-Lhadj and A-K Hamou-Lhadj, “Towards a Compliance Support Framework for Global Software Companies”, *In Proc. of the Software Engineering Conference*, 2007.

[3] D. Cougias, M. Halpern, R. Herold. *Say What You Do: Building a Framework of It Controls, Policies, Standards, and Procedures.* Schaser-Vartan Books, 2007.

[4] D. Vogel, "FDA Regulation of Software for Medical Device Manufacturers", URL: http://pharmaceuticalvalidation.blogspot.com/2010/04/fda-regulation-of-software-for-medical.html

[5] T.D. Breaux, “Legal Requirements Acquisition for the Specification of Legally Compliant Information Systems”, PhD thesis, Computer Science, North Carolina State University, 2009.

[6] REGNET Project: URL: http://eil.stanford.edu/regnet/

[7] OMG GRC-Round Table, URL: http://grc-directory.omg.org/

[8] H. Mehrfard, H. Pirzadeh, A. Hamou-Lhadj, " Investigating the Capability of Agile Processes to Support Life-Science Regulations: The Case of XP and FDA Regulations with a Focus on Human Factor Requirements", H. Mehrfard, H. Pirzadeh, A. Hamou-Lhadj, "Investigating the Capability of Agile Processes to Support Life-Science Regulations: The Case of XP and FDA Medical Devices", In the book series on Studies in Computational Intelligence, Springer Berlin / Heidelberg, 2010.