World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:6, 2016

The Impact of Regulatory Changes on the Development of Mobile Medical Apps

M. McHugh, D. Lillis

***Abstract*—**Mobile applications are being used to perform a wide variety of tasks in day-to-day life, ranging from checking email to controlling your home heating. Application developers have recognized the potential to transform a smart device into a medical device, by using a mobile medical application i.e. a mobile phone or a tablet. When initially conceived these mobile medical applications performed basic functions e.g. BMI calculator, accessing reference material etc.; however, increasing complexity offers clinicians and patients a range of functionality. As this complexity and functionality increases, so too does the potential risk associated with using such an application. Examples include any applications that provide the ability to inflate and deflate blood pressure cuffs, as well as applications that use patient-specific parameters and calculate dosage or create a dosage plan for radiation therapy. If an unapproved mobile medical application is marketed by a medical device organization, then they face significant penalties such as receiving an FDA warning letter to cease the prohibited activity, fines and possibility of facing a criminal conviction. Regulatory bodies have finalized guidance intended for mobile application developers to establish if their applications are subject to regulatory scrutiny. However, regulatory controls appear contradictory with the approaches taken by mobile application developers who generally work with short development cycles and very little documentation and as such, there is the potential to stifle further improvements due to these regulations. The research presented as part of this paper details how by adopting development techniques, such as agile software development, mobile medical application developers can meet regulatory requirements whilst still fostering innovation.

***Keywords*—**Medical, mobile, applications, software Engineering, FDA, standards, regulations, agile.

I. Introduction

IN early 2009, Apple first demonstrated how mobile devices could be used in connection with medical devices. At their annual World Wide Developer Conference they connected blood pressure monitors and blood glucose meters to an IPhone via Bluetooth and cable [1]. In 2014, a report was released which stated that there are over 100,000 Mobile Medical Applications (MMA) available on the two major mobile platforms, IOS and Android. The same report stated that this market was worth $4 billion and that this could potentially rise to as much $26 billion by 2017 [2]. Evidence of this growing popularity can be seen by the major platform providers who are not simply supporting these applications (apps), but are now integrating these apps directly with their operating systems e.g. Google Fit, Health from Apple and Microsoft Health.

MMAs typically meet the definition of being mHealth i.e. "*the use of wireless communication to support efficiency in public health and clinical practice"* [3]. Furthermore, by their very nature, MMAs are deemed as medical device software. Medical device software can be:

- Standalone software;
- Embedded software;
- Software, which transforms a device into a regulated medical device.

To accompany this, research has shown that not only are these apps being developed at a high rate, but also clinicians worldwide are adopting them. In 2015, approximately 500 million people used mobile medical applications [4], [5].

With MMAs becoming increasingly prevalent, regulatory bodies determined that regulations or guidance was needed for app developers to establish if they required regulatory approval prior to being marketed for use [6]. Within the United States (US), the Food and Drug Administration (FDA) regulate medical devices, and within the European Union (EU), medical device regulations are created by the European Council and enforced by notified bodies within each member state.

The first step in regulating MMAs was the release of the FDA Final Rule on Medical Device Data Systems (MDDS). This rule aimed to provide clear guidance as to when software or devices could be deemed as Class I devices and would require the lowest level regulatory scrutiny. However, confusion remained amongst apps developers if their app did not meet the definition of being a MDDS or a MMA. To provide further guidance, the FDA released its draft guidance for mobile medical application developers in 2011 with the final version of this guidance document being released in 2015. The aim of this document was to remove ambiguity surrounding the regulation of MMA.

In the past, app development organizations, who have attempted to innovate and revolutionize the healthcare industry through MMAs, have been stifled by regulations leading to a reluctance to other manufacturers to enter the market [7]. One of the key advantages to mobile applications is that development costs are typically low, as the application is not as complete as a fully-fledged software application [8]. Mobile application manufacturers typically do not produce comprehensive documentation and do not develop their apps in accordance with any defined software development technique. Whilst this approach may be acceptable when developing traditional mobile applications, it is not acceptable

M. McHugh is with the Computer Science at Dublin Institute of Technology (DIT) Dublin, Ireland and is also a member of the Applied Intelligence Research Centre (AIRC) (phone: + 353 (0) 1 402 2808; e-mail: Martin.McHugh@dit.ie).

D. Lillis is with the School of Computing DIT and is also a member of AIRC (e-mail: Deirdre.Lillis@dit.ie).

for developing MMAs, as regulatory bodies require comprehensive documentation as evidence of the safety and efficacy of the application.

Agile software development techniques have been adopted by traditional app development organizations. Agile methods offer reduced development costs, improved time to market and a shorter development lifecycle [9]. However, research has revealed a slow rate of adoption of agile methods by organizations developing software for use in the medical domain [10]. Where agile methods have been adopted for developing software for use in the medical domain, the organizations involved have reported significant benefits [11], [12].

The remainder of this paper is structured as follows: Section II provides background to the approach taken by regulatory bodies with regards to the use of software in a healthcare environment; Section III explains further what a mobile medical application is in terms of how is its viewed by regulatory bodies; Section IV discusses how software development techniques, such as agile software development, can be used to achieve regulatory deliverables whilst fostering innovation when developing MMAs, and Section V provides the conclusions derived from this research.

II. Background

A. Regulations

In 1981, the FDA began to investigate the use of software in healthcare. Initially, the FDA classified medical device software based upon its Draft Software Policy published in 1987 and revised in 1989 [13]. However, the FDA recognised that as the rate of computer and software-based products was growing at an exponential rate, it was not practical to adopt a single "software" policy, which would cover all computer and software based products. Consequently, the draft software policy was withdrawn in January 2005 [14]. As a result, the FDA does not specifically regulate software; rather it regulates devices used in healthcare, which meet the definition of being a medical device. The FDA definition of what constitutes a medical device is outlined in section 201(h) of the Federal Food Drug & Cosmetic (FD&C) Act [15].

> *"an instrument, apparatus, implement, machine, contrivance, implant, in vitro reagent, or other similar or related article, including a component part, or accessory which is:*
> - *recognized in the official National Formulary, or the United States Pharmacopoeia, or any supplement to them,*
> - *intended for use in the diagnosis of disease or other conditions, or in the cure, mitigation, treatment, or prevention of disease, in man or other animals, or*
> - *intended to affect the structure or any function of the body of man or other animals, and which does not achieve its primary intended purposes through chemical action within or on the body of man or other animals, and which is not dependent upon being metabolized for the achievement of any of its primary intended purposes."*

It can be seen from the definition provided that if software performs any of the functions outlined in the definition of a medical device, then it becomes subject to scrutiny by the FDA.

1. Safety Classification

All medical devices marketed for use within the US must receive a safety classification. This classification is determined based upon the potential risk a medical device poses on patients, clinicians or third parties. The three classifications are, Class I Low Risk, Class II Medium Risk and Class III High Risk. All medical devices initially receive a Class III safety classification unless they meet the definition of being in a device category with a lower classification or until they are reclassified by the FDA.

2. 21 CFR 820 Quality Systems Regulations

All medical devices marketed for use within the US regardless of device safety classification, must provide evidence of adoption of a Quality Management System (QMS), such as in accordance with 21 CFR 820 Quality Systems Regulations (QSR) [16] and the FDA Design Control Guidance for Medical Device Manufacturers [17]. Of note within the QSR is Subpart C – Design Controls, which provides information as to which processes must be adhered to when developing regulatory compliant software. These include:

- Design & Development Planning (Specifications);
- Design Output (Coding);
- Design Review;
- Design Verification (Was the Product Built Right?);
- Design Validation (Was the Right Product Built?).

The primary objective of the QSR is to ensure the safe and reliable performance of a medical device. A device is deemed safe if it does not cause harm to a patient, clinician or third party and it is deemed reliable if it performs the desired function each and every time it is used.

3. 21 CFR 880 Medical Devices; Medical Device Data Systems Final Rule

Prior to April 16, 2011, devices that now meet the current definition of being a MDDS were classified as either a Class III device, or assumed the safety classification of the parent medical device to which they were connected, although the FDA had been operating under their discretionary enforcement policy and therefore was not enforcing the Class III requirements on all MDDS. However, on April 16, 2011, a FDA rule became effective which classified a MDDS device as Class I, 510 (k) exempt - medical device [14]. This ruling came three years after the proposed ruling was issued on February 8, 2008. This final classification modifies FDA 21 C.F.R § 880.6310 and describes a MDDS as being:

> *"software, electronic, or electrical hardware such as a physical communications medium (including wireless hardware), modems, interfaces and communications protocol."*

The FDA provided the following definition of what constitutes a MDDS:

"A device that is intended to provide one or more of the following uses, without controlling or altering the functions or parameters of any connected medical devices:

(i) The electronic transfer of medical device data;
(ii) The electronic storage of medical device data;
(iii) The electronic conversion of medical device data from one format to another format in accordance with a pre-set specification; or
(iv) The electronic display of medical device data."

There is however, an exception to this rule. If software exclusively performs one or more of the functions outlined in the definition of a MDDS and is used for active patient monitoring, then it cannot be considered a MDDS and must be considered an accessory or medical device in its own right.

B. Standards

In November 1997, the FDA signed into law the Modernization Act, known as the Food and Drug Administration Modernization Act (FDAMA) [18]. A key element of the FDAMA is the advocating of the use of standards in the design review process. To support the FDAMA, the FDA published in the Federal Register, a list of standards to which medical device manufacturers could declare conformity. A key objective of the FDAMA was to reduce the burden on both the FDA and medical device manufacturers by reducing the regulatory obstacle to entry to international and domestic medical device markets. When the FDAMA was signed into law, the Centre for Devices and Radiological Health (CDRH) established Standards Technology Groups (STG), one of which had a specific focus on software. A STG is responsible for software categorized as follows:

- General process standards, which are technology independent;
- General process standards, which are technology dependent;
- Specific process implementations.

A number of standards are included on the Federal Register list of standards; of most significance with regards to medical device software development is IEC 62304:2006 Medical Device – Software Life Cycle Processes [19]. Also of significance to medical device software and all types of medical device is ISO 14971:2012 Application of Risk Management to Medical Devices [20]. In the EU, ISO 13485:2012 [21] Medical Devices – Quality Management Systems – Requirements for Regulatory Purposes is central to the development of regulatory compliant software. However, prior to March 2012, medical device companies wishing to market a medical device within the US were required to provide evidence of adherence to the FDA QSR regulations. Therefore, if a medical device manufacturer was developing a medical device for use in the EU and the US, they needed to conform to both of the quality management system guidelines. In March 2012, the FDA commenced a pilot program offering device manufacturers the option of submitting their quality system audits, which are compliant with ISO/IEC 13485:2012, to the FDA's Centre for Devices and Radiological Health (CDRH) or Centre for Biologics Evaluation and Research (CBER) [22]. This is seen as a step forward in the FDA's plan to create a partnership with Health Canada, which would result in a single audit program for both the US and Canada.

1. IEC 62304:2006 Medical Device – Software Life Cycle Processes

As medical devices are safety critical, manufacturers are recommended to follow current international standards during development. Adherence to these standards is not mandatory, but it is recommended in order to achieve regulatory approval. Adherence to the standards demonstrates the manufacturer's ability to follow defined development procedures and to perform the required risk management activities [23]. If a manufacturer chooses not to adhere to these standards, they must provide a sufficient explanation as to why and they must demonstrate that the alternative method chosen is equally valid. Within the US, it is the responsibility of the FDA to ensure compliance with these standards. FDA auditors and Inspectors perform these compliance checks.

IEC 62304:2006 is the current software development standard followed by medical device software developers. The current version of IEC 62304 was released in 2006. IEC 62304 is derived from ISO/IEC 12207:1995 Software Lifecycle Processes [24], AMD 1:2002 [25] and 2:2004 [26]. ISO 12207:1995 is not domain specific but it is seen as being comprehensive in its approach which is reflected in the number of standards that utilize the core principles of ISO 12207:1995, AMD 1 and AMD 2 as their foundation, such as ISO/IEC 15504-5:2006. IEC 62304 is domain specific and is tailored to suit the specific requirements of the medical device software development industry. IEC 62304 is a software development standard that provides end-to-end guidance in the development of the software component of a medical device. However, it hands off system activities such as Requirements Elicitation and Validation, to its aligned standards which include ISO 13485:2003 Medical Devices – Quality Management Systems [27], ISO 14971:2007 Medical Devices – Application of Risk [28] and ISO/IEC 15288:2008 – Systems and Software engineering – System lifecycle processes [29]. IEC 62304 is a harmonized standard under the MDD [30] and is a FDA consensus standard [31].

IEC 62304 makes provision for the application of risk to software. IEC 62304 applies a classification system to software components similar to that of ISO 14971 Clause 4.4.5 and 6.1. The safety classification is as follows:

- Class A – No injury or damage to health possible;
- Class B – Non-serious injury is possible;
- Class C – Death or serious injury is possible.

The risk classification applied to an item of software is determined by the amount of potential risk the medical device places upon the patient, clinician or third party. With IEC 62304, the overall software component assumes the safety classification of the software element that poses the most risk.

However, IEC 62304 does make allowance for software components to be segregated into individual software elements with each element receiving its own safety classification.

2. ISO 13485:2012 Medical Devices – Quality Management Systems – Requirements for Regulatory Purposes

ISO 13485:2012 was published in 2012, which forms the basis for the development of a quality management system when developing medical device software. ISO 13485 is derived from ISO 9001; however, ISO 13485 is tailored to include elements specific to the development of medical devices. Additionally, ISO 9001 requires a device manufacturer to perform continuous improvement, while ISO 13485 only requires a manufacturer to implement and maintain a quality management system. As previously discussed, the FDA now accepts quality management audits performed in accordance with ISO 13485.

3. ISO 14971:2012 – Application of Risk Management to Medical Devices

ISO 14971 was first released in 2000, with the second edition released in 2007 and the latest version released in 2012. ISO 14971 is an FDA consensus standard. It specifies the procedures and activities for identifying hazards in medical devices and accessories to medical devices, including software. ISO 14971 provides guidelines to medical device manufacturers on the preparation of a plan to prepare for risk management activities. The plan should contain:

- The scope of the plan;
- A verification plan;
- Allocation of responsibilities;
- Requirements for review of risk management activities;
- Requirements for collecting and reviewing production and post-production information;
- Criteria for risk acceptability.

ISO 14971 is not specific to the development of medical device software. As a result, it can be difficult to apply it to the development of medical device software. Consequently, IEC produced a Technical Report (TR) providing guidance to medical device manufacturers on applying ISO 14971. This guidance document is known as IEC/TR 80002-1:2009 – Part 1: Guidance on the application of ISO 14971 to medical device software [32]. IEC/TR 80002-1:2009 follows the same structure as ISO 14971, making it easier to follow for those familiar with ISO 14971's structure.

III. Mobile Medical Applications

A. When is an App a Mobile Medical App?

A mobile application is defined as standalone software that exists on a smart device such as a mobile phone or tablet [8]. Table I shows a number of key terms and definitions relevant to mobile applications.

The FDA Mobile Medical Applications, Guidance for Industry and Food and Drug Administration Staff [33] defines a mobile medical application as:

"a mobile medical application is a mobile app that meets the definition of a device in section 201(h) of the Federal Food, Drug, and Cosmetic Act; and either is intended:

- *To be used as an accessory to a regulated medical device; or*
- *To transform a mobile platform into a regulated medical device."*

TABLE I
Key Terms and Definitions [8]

Term	Definition
Native	Software that comes pre-installed on a mobile device such as Google Fit.
Downloadable	Software that is not pre-installed on a device but can be downloaded and installed from another sources.
Web-Based	An application that is accessed via a mobile device however no installation or download occurs to the mobile device.
Mobile Application Store	An online portal which facilities the searching for, and downloading of downloadable mobile application e.g. App store or Google Play.
Mobile Device OS	The primary operating system, which resides on a mobile device e.g. IOS or Android.

The FDA recognized that the definition could cause confusion and as such, they went on further to explain that the intended use of the mobile application would determine whether or not it meets the definition of being a "device". For example, if a mobile application developer makes an app that turns on a mobile device's flash for the purpose of a torch for general use, then this app is not defined as being a medical device. However, if the app developer through marketing and labeling intended the app for use by clinicians to assist with their daily tasks, then this application would meet the definition of being a "device". In instances such as this, the labeling and marketing would establish the intended use, as the app could potentially be used within multiple domains. However, where apps have been developed with a clear medical focus but have not been labeled and marketed for medical use, then this app would still meet the definition of being a device as no ambiguity surrounds the intended use. For example, if an app developer created an app that helped analyze ECG results, then regardless of labeling, this app would be subject to regulatory controls. Prior to the FDA releasing its draft guidance on mobile medical applications, app developers were avoiding any form of regulatory control by labeling their apps as lifestyle apps. This loophole has now been closed.

B. Who is a Mobile Medical Application Manufacturer?

In other domains it may be clear who the developer of a specific software product is i.e. the software development organization producing a software package would be deemed the manufacturer. However, in line with other medical device regulations, confusion initially arose as to who was defined as being a mobile medical application manufacturer. In the production of other forms of medical devices such as infusion pumps and CT scanners, FDA 21 CFR Parts 803 [34], 806 [35], 807 [36] and 820 [16], defined the definition of the manufacturer. This definition of a manufacturer includes anyone who:

- Develops specification;

- Designs;
- Labels;
- Creates a software system or application for a regulated medical device.

With the release of the FDA final guidance on mobile medical applications, the ambiguity surrounding who is defined as a manufacturer has now been removed. Certainty has also been provided in that distributors of MMAs do not meet the definition of being manufacturers and as such avoid regulatory scrutiny i.e. the Apple App Store and Google Play Store do not need to apply for regulatory approval prior to distributing MMAs.

TABLE II
CATEGORIES OF MMAS BY THE FDA [4]

Applications Functionality	Example	Consideration
As an extension of approved medical device including displaying, storing, analyzing, or transmitting patient specific data	Display of medical images X-Rays and MRI, graphic data such as EEG waveforms, bedside monitors	High Risk – good resolution of the screen is extremely important in certain cases like X-Ray/MRI as lower resolution may affect clinical decision negatively
Applications that convert a mobile platform into a medical device	Converting phone/smart watches into urine analyzers or glucometers. Attachment of transducers to make stethoscopes, spirometers	High Risk – readings may directly affect the clinical decisions therefore the apps need to be extremely accurate.
Applications/Websites diagnosing & recommending treatment options on the basis of patient specific input	Prognosis of the disease, treatment options, dosage calculators	Medium Risk – The geographic region is very important. If a drug is not available over the counter and patients need a prescription then its low risk.
Apps for general health applications & education purposes	BMI Calculators, heart rate monitors, thermometers, medication reminders	Low Risk – Marketing claims are critical for products to be placed in general health benefits category, which is very low risk, most health applications for mass public consumption are likely to fall under this category

C. Mobile Medical Application Safety Classification

As with all software or devices used in connection with patient care, once a mobile application meets the definition of being a mobile medical application it must be classified as Class I, Class II or Class III. As discussed previously, all mobile medical applications initially are classified as Class I devices if they meet the definition of being an MDDS, or as a Class III device until reclassified by the FDA.

The FDA, as part of its guidance has also covered guidance as to MMAs, which it intends to exercise enforcement discretion[2]. Examples include apps which:

- Help patients self-manage their disease of conditions;
- Provide patients with simple tools to organize and track their health information;
- Provide easy access to information related to patients' health;
- Automate simple tasks for health care providers i.e. MDDS.

[2] *The term enforcement discretion means that even if a mobile application may meet the definition of being a medical device, the FDA can choose not to enforce our requirements because we determined that the risk to patients is low* – Bakul Patel, Senior Policy Advisor to the center director – CDRH and FDA [37]

IV. DEVELOPING MOBILE MEDICAL APPLICATIONS

A. Developing Regulatory Compliant Software

Despite not dictating a Software Development Life Cycle (SDLC) to follow, medical device software development organizations typically follow the V-Model [38]. It was first presented in 1991 at the NCOSE symposium [39] and is a variation on a SDLC which Royce presented which later became known as the Waterfall Model [40]. The V-Model identifies that there are different types of testing such as modular testing and integration testing [41]. The V-Model shows the relationship between the two sides of the development process. This relationship is used to determine whether the stage has been completed successfully. If a problem occurs during the verification or validation of any one stage, then the opposite stage on the "V" must be revisited and if necessary, reiterated [42]. Essentially, the testing of a product is planned in parallel with the corresponding phase of development. This method of developing software eases the process of achieving traceability. The FDA mandates that traceability be an integral part of a development process [43]. Therefore, the V-Model is perceived to be the "best fit" with the regulatory requirements. While it may be the best fit, in practice the V-Model presents the same problems that are associated with utilizing any sequential plan driven SDLC. Royce, who presented the Waterfall model, stated that there are inherent problems associated with following a sequential lifecycle [40]. For example, as requirements are fixed at such an early stage, it can be very difficult to introduce a change in requirements once the project is underway. Furthermore, it can be very difficult to capture all of the requirements at such an early stage of a project [44]. In addition to this, any changes introduced once a project is underway can create cost and budget overruns [45].

B. Agile Software Development

Recognizing the inadequacies associated with plan-driven approaches, a shift has occurred toward a more flexible or agile approach to software development. Agile software development was first formalized in 2001 and since then has gained greater acceptance in the software development industry. This is evident in a large scale survey of software development organizations, conducted in 2013, which identified that 88% organizations stated that they were following an agile approach [46]. This is an increase from 84% in 2012 [47] and 80% in 2011 [48].

The principles of agile software development originate from the "Agile Manifesto" [49]. In February 2001, a meeting was held at The Lodge at Snowbird ski resort in Utah. At this meeting, 17 people met including Kent Beck, Alistair

Cockburn, and Robert C. Martin, all very experienced in the field of software engineering, to discuss software development methodologies. As a result of this meeting, the agile software development alliance emerged. The agile alliance determined the priorities of a development project, as shown in Table III:

TABLE III
AGILE SOFTWARE DEVELOPMENT VALUES

Individuals and interactions *over* processes and tools;
Working software *over* comprehensive documentation;
Customer collaboration *over* contract negotiation;
Responding to change *over* following a plan.

In essence, agile principles place a greater importance on the human component of a development project, rather than on utilising a rigid plan i.e. people over processes [50].

Cursory reading of these values would appear to suggest that agile methods are contradictory to regulatory requirements as none of the regulatory deliverables such as documentation are produced. Further examination of the agile values identified that the statements on the left are deemed of greater importance than those on the right in an agile project, however they do not replace the items on the right. For example, as identified by Robert Martin [50], one of the authors of the Agile Manifesto states "Produce no document unless it's immediate and significant" demonstrating that as long as there is value to be obtained by the items on the right, they should still be produced.

Regulatory requirements have been put in place to ensure that software produced for use in connection with patient care is of the highest quality. Furthermore, quality is one of the main aims of the agile software development. In agile, software development, planning, requirements definition, design, testing and validation are all performed, however, they are performed over several increments which are designed to give feedback early in a software project. This approach allows for greater clarity and control of a software development process, more so than traditional plan driven approaches such as the Waterfall of V-Model. This feedback loop can be modified to incorporate risk management, human factors, and verification and validation that meet the FDA's quality system regulations.

Another example as to how agile methods may be incompatible in the medical device domain is the difficulty associated with incremental development. In non-regulated software development adopting agile approaches, the software is developed partially, examined and if necessary, reiterated. This examination can come in the form of alpha or beta testing or possibly release to end users with the subsequent elements being updated based on the feedback obtained. In the medical domain and other safety-critical domains, it is not possible to release software into a live environment without adequate testing and regulatory approval.

AAMI TIR45:2012

In October 2012, the Association for the Advancement of Medical Instrumentation (AAMI) released a Technical Information Report (TIR) known as AAMI TIR 45:2012 Guidance on the use of agile practices in the development of medical device software [51]. The committee that developed the TIR consisted of industry experts and FDA staff. AAMI recognized the shift in the generic software development industry towards more agile practices and the evidence presented from successful adoption of agile practices in medical device software development organizations. However, they identified that the available information with regards to the adoption of agile practices when developing medical device software was hard to understand and the objective of the TIR is to provide clear guidance of which practices of agile software development are suited to the development of medical device software. The TIR also provides recommendations for complying with international standards and FDA guidance documents when using agile practices to develop medical device software.

The TIR focuses on a number of areas in which agile software development practices are suited when developing medical device software. These areas include:

- Planning;
- Team Structure and Collaboration;
- Product Definition and Requirements Documentation;
- Software Architecture;
- Detailed Design;
- Implementation and Unit Verification;
- Integration and Integration Testing;
- Software System Testing;
- Software Release;
- Configuration Management and Change Management;
- Corrective and Preventative Action.

The TIR successfully maps practices performed as part of agile software development techniques to each of these stages of development. Whilst the TIR can be seen as useful when developing medical device software, two issues can potentially arise. Firstly, the TIR maps agile practices to IEC 62304; IEC 62304 only provides guidance for the development of the software portion of a medical device system and therefore, it could be difficult to apply the TIR to the development of standalone software. Secondly, the TIR only provides high-level guidance as to specific agile practices that can be used when developing medical device software. Many more agile practices exist which could potentially be used in the development of medical device software, but are not included in the TIR. This TIR is not a hands-on approach on agile methods. It provides a good discussion on what can be done or what can't be done with agile methods, and it serves to reassure people who are skeptical about agile methods. Additionally, while this TIR may not be comprehensive in its approach, it does serve as evidence of the changing attitude of the FDA with regards the use of agile software development approaches when developing regulatory compliant software.

C. Can Agile Approaches Be Used in Practice?

Rasmussen et al. [11] detailed the successful implementation of agile practices within Abbott Diagnostics, the organization recognized the need to move away from a plan-driven approach. In this implementation, Abbott

completed two projects side-by-side, one in accordance with agile methods and the other in accordance with a plan-driven approach. While both projects were not the same size, the organization identified that the project completed in accordance with agile methods made a cost saving of between 35% and 50% when compared to the plan-driven project.

Rottier and Rodrigues [12] detailed the implementation of an agile approach within Cochlear. As with Abbott Diagnostics, Cochlear wished to streamline their development process by moving away from a plan-driven approach to a more agile one. However, they quickly identified that it was not possible to wholly adopt a single agile method, such as Scrum or XP on its own, as no single agile method provides sufficient guidance of each of the stages, which are necessary when developing medical device software. This supports the findings of Vogel [52] and Turk et al. [53], who also identified that no single agile method is sufficiently comprehensive for use when developing medical/safety critical software. Within Cochlear, it was identified that combining an agile method such as Scrum with a plan-driven SDLC such as the V-Model, sufficient guidance for the development of regulatory compliant software is provided.

Spence [54] discussed the implementation of Scrum within Medtronic. Spence identified that it is not practical to follow a rigid plan-driven approach when developing medical device software, as it is not possible to fully complete one stage of development before moving on to the next, whilst ruling out the need to revisit a stage. The research conducted by Spence is related more to the organizational challenges associated with implementing agile in a medical device software development organization. Further to this, Weyrauch [55] published research on the adoption of agile practices within Medtronic. He builds further on the information presented by Spence, however, the detail he presented remained closer to the organizational impact and accommodation of agile methods, rather than the impact agile practices had on a software development project.

Weiguo and Xiaomin [56] presented a SDLC, which incorporates practices with a plan-driven approach and was implemented on a medical device software development project. Unfortunately, the information presented by the authors is very sparse and they do not provide enough guidance as to how their tailored SDLC was implemented should an organization wish to adopt their SDLC. However, they do outline that rather than wholly adopting a single agile method, they retained the V-Model/plan-driven approach to produce the necessary regulatory deliverables.

While the detail included as part of each of these implementations is sparse, commonalities can be identified. Each of the organizations initially examined the possibility of wholly adopting a single agile method such as Scrum or XP. They soon realized this was not possible and as such they integrated selected agile practices with their traditional plan-driven approach. Furthermore, the selected practices typically originated from either the Scrum or XP approaches.

V. Conclusions

The FDA does not regulate specific forms of medical devices; rather they regulate all devices that are intended for use in connection with patient care. While the regulations attempt to provide clear information to device manufacturers, ambiguity can arise when they try to apply the regulations to specific medical device sectors such as medical device software. As a result, the FDA has released a number of guidance documents to help medical device software manufacturers to navigate the regulatory process. The latest guidance document released by the FDA is intended for MMA manufacturers and FDA staff. This document details when a mobile application is deemed a mobile medical application and if so, the necessary steps, which must be taken in order to achieve regulatory approval. These steps increase the overhead associated with developing such apps and as such, there is the potential to deter mobile medical application manufacturers from entering the market, which could ultimately lead to a reduction in competition and advancements. This paper discussed how by adopting agile software development techniques MMA manufacturers could continue to develop apps which meet regulatory approval, whilst not sacrificing the approach which they may be accustomed i.e. producing little documentation and fast development cycles. To achieve this, MMA manufacturers are advised to follow a hybrid software development approach. This hybrid approach would involve following the V-Model. This will assist in producing regulatory deliverables along with adopting agile practices, which will assist in promoting development and innovation.

References

[1] B. Dolan. (2010). *Timeline: The iPhone as medical tool.* Available: http://mobihealthnews.com/2580/timeline-the-iphone-as-medical-tool/ (Accessed 03/11/15)

[2] Research2Guidance, "Fourth annual study on mHealth app publishing," Continua, Research2Guidance2014.

[3] J. G. Kahn, J. S. Yang, and J. S. Kahn, "'Mobile'health needs and opportunities in developing countries," *Health Affairs,* vol. 29, pp. 252-258, 2010.

[4] A. K. Yetisen, J. L. Martinez-Hurtado, F. da Cruz Vasconcellos, M. C. E. Simsekler, M. S. Akram, and C. R. Lowe, "The regulation of mobile medical applications," *Lab on a Chip,* vol. 14, pp. 833-840, 2014.

[5] K. F. B. Payne, H. Wharrad, and K. Watts, "Smartphone and medical related App use among medical students and junior doctors in the United Kingdom (UK): a regional survey," *BMC medical informatics and decision making,* vol. 12, p. 121, 2012.

[6] O. I. Franko, "How Helpful Are Mobile Healthcar Apps?" *AAOS Now*, vol. 7, 2013.

[7] FDA. (2013, 27/10/15). *Letter to Biosense Technologies Private Limited concerning the uChek Urine Analyzer*.

[8] A. J. Barton, "The regulation of mobile health applications," *BMC medicine,* vol. 10, p. 46, 2012.

[9] D. J. Reifer, "How good are agile methods?" *Software, IEEE,* vol. 19, pp. 16-18, 2002.

[10] M. McHugh, O. Cawley, F. McCaffery, I. Richardson, and X. Wang, "An Agile V-Model for Medical Device Software Development to Overcome the Challenges with Plan-Driven Software Development Lifecycles," presented at the Software Engineering in Healthcare (SEHC) workshop at the 35th International Confernence on Software Engineering (ICSE), San Francisco CA, 2013.

[11] R. Rasmussen, T. Hughes, J. R. Jenks, and J. Skach, "Adopting Agile in an FDA Regulated Environment," presented at the Agile Conference, 2009 AGILE '09 Chicago, IL 2009.

[12] P. A. Rottier and V. Rodrigues, "Agile Development in a Medical Device Company," presented at the Proceedings of the 11th AGILE Conference. AGILE '08., Toronto, 2008.
[13] P. T. H. Kim, "FDA and the regulation of medical software," presented at the Sixth Annual IEEE Symposium on Computer-Based Medical Systems, Michigan, 1993.
[14] FDA, "21 CFR Part 880 Medical Devices; Medical Device Data Systems Final Rule.," *Federal Register* vol. 76, pp. 8637 - 8649, 2011.
[15] *Federal Food, Drug, and Cosmetic Act, 21 U.S.C. §§ 301*, 2004.
[16] FDA, "Title 21--Food and Drugs Chapter I --Food and Drug Administration Department of Health and Human Services subchapter h--Medical Devices part 820 Quality System Regulation," U.S. Department of Health and Human Services, Maryland2007.
[17] FDA, "FDA Design Control Guidance for Medical Device Manufacturers," Centre for Devices and Radiological Health, Maryland1997.
[18] *FDAMA §114; 21 USC 352(a)*. 1997.
[19] AAMI, "ANSI/AAMI/IEC 62304, Medical device Software - Software life cycle processes," ed. Association for the Advancement of Medical Instrumentation, 2006.
[20] ISO, "ISO/IEC 14971:2012 Medical devices -- Application of risk management to medical devices," International Organisation for Standards, Geneva, Switzerland 2012.
[21] ISO, "*ISO/IEC 13485:2012 Medical devices -- Quality management systems -- Requirements for regulatory purposes*," ed. International Organisation for Standards - Geneva, Switzerland, 2012.
[22] *Guidance for Industry, Third parties annd Food and Drug Administration Staff, Medical Device ISO 13485:2003 Voluntary Audit Report Submission Program*, 2012.
[23] S. Miura, "Industry Viewpoint on software," presented at the 11th Conference of the Global Harmonization Task Force, Washington, 2007.
[24] ISO, "ISO/IEC 12207 Information technology -- Software life cycle processes," International Organisation for Standards, Geneva, Switzerland1995.
[25] ISO, "ISO 12207:1995/AMD1:2002 Information technology - Software life cycle processes AMENDMENT 1," International Standards Organisation, Geneva Switzerland2002.
[26] ISO, "ISO 12207:1995/AMD2:2004 Information technology - Software life cycle processes AMENDMENT 2," International Standards Organisation, Geneva Switzerland2004.
[27] ISO, "*ISO/IEC 13485:2003 Medical devices -- Quality management systems -- Requirements for regulatory purposes*," ed. International Organisation for Standards - Geneva, Switzerland, 2003.
[28] ISO, "ISO/IEC 14971:2007 Medical devices -- Application of risk management to medical devices," International Organisation for Standards, Geneva, Switzerland 2007.
[29] ISO, "ISO/IEC 15288:2008 Systems and software engineering -- System life cycle processes," International Organisation for Standards, Geneva, Switzerland2008.
[30] European Council, "Commission communication in the framework of the implementation of the Council Directive 93/42/EEC of 14 June 1993 concerning medical devices (Publication of titles and references of harmonised standards under the directive)," European Council, EU Bookshop2010.
[31] C. Gerber. (2008). *Introduction into IEC 62304 Software life cycle for medical devices*. Available: http://www.spiq.com/abs/JF200809IEC62304 SPIQ Rev004.pdf (Accessed 10/10/2015)
[32] AAMI, "AAMI IEC TIR 80002-1:2009 Medical Device Software - Part 1: Guidance on the Application of ISO 14971 To Medical Device Software," Ed, 2009.
[33] FDA, "Guidance for Industry and Food and Drug Administration - Mobile Device Applications," Centre for Devices and Radiological Health, Maryland2015.
[34] FDA, "21 CFR Part 803 Medical Device Reporting," *Federal Register* vol. 79, pp. 8846-8855, 2014.
[35] FDA, "21 CFR Part 806 Medical Devices; Reports of Corrections and Removals," *Federal Register* vol. 8, 2015.
[36] FDA, "21 CFR Part 807 Establishment Registration and Device Listing for Manufacturers and Initial Importers of Devices," *Federal Register* vol. 8, 2010.
[37] P. Bakul, "The Regulatory Perspective: Q&A with FDA's Bakul Patel," *Biomedical Instrumentation & Technology,* vol. 46, pp. 84-86, 2012.
[38] F. McCaffery, D. McFall, P. Donnelly, and F. G. Wilkie, "Risk Management Process Improvement for the medical device industry," presented at the Conference on Software Development (SWDC-REK-2005) Iceland, 2005.
[39] K. Forsberg and H. Mooz, "The Relationship of Systems Engineering to the Project Cycle," presented at the First Annual Symposium of the national Council on Systems Engineering (NCOSE), Chattanooga, Tennessee, 1991.
[40] W. Royce, "Managing the Development of Large Software Systems," presented at the Proceedings of IEEE WESCON, 1970.
[41] P. E. Rook, "Controlling software projects," *IEEE Software Engineering Journal,* vol. 1, p. 7, 1986.
[42] S. L. Pfleeger and J. M. Atlee, *Software Engineering: Theory and Practice*. New Jersey: Pearson Higher Education, 2001.
[43] V. Casey and F. McCaffery, "Med-Trace: Traceability Assessment Method for Medical Device Software Development.," presented at the European Systems & Software Process Improvement and Innovation Conference, (EuroSPI). Roskilde, Denmark, 2011.
[44] J. Cadle and D. Yeates, *Project Management for Information Systems*: Pearson Education, 2008.
[45] N. M. A. Munassar and A. Govardhan, "A Comparison between Five Models of Software Engineering," *IJCSI International Journal of Computer Science Issues,* vol. 7, pp. 94-101, 2010.
[46] VersionOne, "8th Annual State of Agile Survey: The State of Agile Development," 2013.
[47] VersionOne, "7th Annual State of Agile Survey: The State of Agile Development," 2012.
[48] VersionOne, "6th Annual State of Agile Survey: The State of Agile Development," 2011.
[49] Beck et al. (2001). *The Agile Manifesto*. Available: http://agilemanifesto.org/ (Accessed 7/8/2015)
[50] R. C. Martin, *Agile Software Development - Principles, Patterns and Practices*. Upper Saddle River, NJ: Prentice Hall, 2003.
[51] AAMI, "AAMI TIR45:2012 -- Guidance on the use of agile practices in the development of medical device software," Association for the Advancement of Medical Instrumentation, Arlington VA2012.
[52] D. Vogel, "Agile Methods: Most are not ready for prime time in medical device software design and development," *DesignFax Online,* vol. 2006, 2006.
[53] D. Turk, R. France, and B. Rumpe, "Limitations of Agile Software Processes," presented at the Third International Conference on Extreme Programming and flexible processes in Software Engineering (XP2002), Alghero, Sardinia, 2002.
[54] J. W. Spence, "There has to be a better way! [software development]," presented at the Proceedings to Agile Conference, 2005. Denver, 2005.
[55] K. Weyrauch, "What Are We Arguing About? A Framework for Defining Agile in our Organization," presented at the Proceedings of the conference on AGILE 2006, 2006.
[56] L. Weiguo and F. Xiaomin, "Software Development Practice for FDA-Compliant Medical Devices," presented at the International Joint Conference on Computational Sciences and Optimization, 2009. Sanya, Hainan, 2009.