QUALITY ASSURANCE FOR DELIVERING MOBILE APPLICATIONS THAT ENABLE HEALTHIER AND MORE PRODUCTIVE LIVES

A Thesis

Presented

to the Faculty of

California State University Dominguez Hills

In Partial Fulfillment

of the Requirements for the Degree

Masters of Science

in

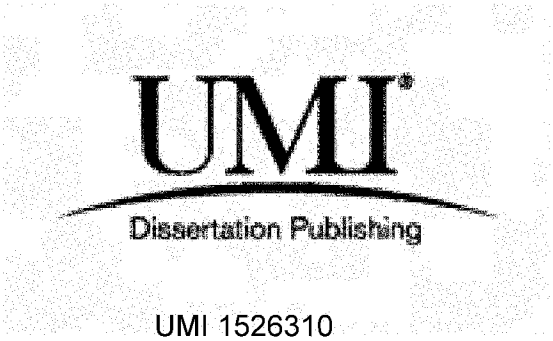
Quality Assurance

---

by

Habteab Yemane

Fall 2014



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

ACKNOWLEDGMENTS

First, I would like to acknowledge my committee members for their input during the course of my research. They played an invaluable role in helping me to complete this piece in a timely, efficient, and appropriate manner. Specifically, I would like to acknowledge Professor Kim Niles for his heartfelt encouragement throughout the completing of my thesis. Additionally, I would like to recognize Professor Keith Fulton for his timely and knowledgeable input regarding my writing style and grammar. Finally, I would like to extend a special thanks to Professor Robert Spencer for his thorough and knowledgeable input regarding the content of my paper.

Secondly, I would like to thank the Graduate Office and related personnel for their guidance, support, and feedback throughout the publishing of this thesis.

Finally, my friends and family have been extremely supportive throughout this process. Specifically, I thank Adam Sunderland for his support, encouragement, and editing throughout the writing of this thesis. In addition, I thank David and Mary Fisher for their love and support. Likewise, I extend a sincere heartfelt thank you to my parents who have always supported me so much. And most important, I would like to extend a loving thank you to my wife, who not only supported me throughout this process, but also gave birth to our beautiful and wonderful daughter, Hannah, in the middle of the writing of this thesis.

TABLE OF CONTENTS

PAGE

COPYRIGHT PAGE....ii

APPROVAL PAGE....iii

ACKNOWLEDGMENTS....iv

TABLE OF CONTENTS....v

LIST OF TABLES....vii

LIST OF FIGURES....viii

ABSTRACT....ix

CHAPTER

1. INTRODUCTION....1
   - Background....1
   - Statement of the Problem....6
   - Purpose of the Study....8
   - Theoretical Bases and Organization....10
   - Limitations of the Study....12
   - Definition of Terms....13

2. REVIEW OF LITERATURE....19

   - Food and Drug Administration Regulatory Documents....19
     - The Quality System Regulation....20
     - Mobile Medical Application Regulations....21
     - Quality System Requirements....21
     - Analysis of Regulatory Requirements....24
   - Mobile Medical Application Market....25
     - Multiple Platforms and Devices....26
     - Device Network Speed and Security....27
     - Unparalleled Customer Quality Requirements....27
     - Short Product Development Cycles....29
     - Testing Difficulty....29
   - The Waterfall Model....30
     - The Waterfall Phases....31
     - Strengths of the Waterfall Model....33

CHAPTER PAGE

Shortcomings of the Waterfall Model 34
Agile Methodologies 35
Agile-Based Quality Systems 38
Shortcomings of the Agile Methodologies 40
Promises of the Agile Methodologies 41

3. METHODOLOGY 46

Quality System for Mobile Medical Application 46
Incremental and Iterative Development Process 48
Agile Augmented Waterfall Phases 50
Fulfilling Market Requirements 57
Multiple Platforms and Devices 57
Device Network Speed and Security. 58
Unparalleled Customer Quality Requirements 59
Short Product Development Cycles 59
Testing Difficulty 61
Case Study: Applying the Agile augmented Waterfall Quality System to the iOxyMonitor 62
iOxyMonitor Requirements Phase 63
IOxyMonitor Design Phase 69
iOxyMonitor Implementation and Testing Phase 72
iOxyMonitor Deployment Phase 76
Rationalizing the Use of the Agile Augmented Waterfall Quality System through its Application to the iOxyMonitor 78

4. RESULTS AND DISCUSSION 80

5. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS 83

Summary 83
Contributions 84
Recommendations 85

REFERENCES 89

LIST OF TABLES

PAGE

1. Waterfall Approach Augmented Through the Application of Agile Practices 47

2. Agile Practices for Addressing the Need to Design for Multiple and Changing Platforms and Devices 58

3. Agile Practices for Addressing the Need to Design for Variations in Network Capability and Security 59

4. Agile Practices for Addressing the Need to Meet Demands for Product Quality 60

5. Agile Practices for Facilitating Shorter Product Development Lifecycles 60

6. Agile Practices for Addressing Testing Difficulty 62

7. Sample of User Stories Developed for the iOxyMonitor 65

8. iOxyMonitor Project Vision Statement 65

9. iOxyMonitor Product Backlog and Requirements Prioritization Results 68

LIST OF FIGURES

PAGE

1. The incremental and iterative cycle of the augmented Waterfall approach 49

2. The requirements phase of the Agile augmented Waterfall model 51

3. The design phase of the Agile augmented Waterfall model 53

4. The implementation and testing phase of the Agile augmented Waterfall model 55

5. The deployment phase of the Agile augmented Waterfall model 56

6. Translation of design inputs into design outputs during the design phase 69

7. Pictures of CRC cards used during the design phase for the iOxyMonitor 70

8. The project team's mapping of a sub-system with CRC cards 71

9. Application of Agile tools and practices to the implementation and testing phase of the iOxyMonitor 74

ABSTRACT

Mobile medical applications are emerging as a significant part of the healthcare industry. Despite this popularity, mobile medical application development has proven difficult due to several factors. Moreover, the U.S. Food and Drug Administration's (FDA) 2013 publishing of strict regulatory requirements has further complicated development efforts.

Consequently, this thesis addresses the market need to design mobile medical applications in a manner that simultaneously ensures conformance to regulatory requirements and facilitates commercial success. Thoroughly analyzed as a solution is the integration of Agile tools, practices, and principles with the Waterfall model for software development to develop a quality system that delivers safe and effective mobile medical applications in a manner that promotes commercial success.

Specifically, this thesis provides an Agile augmented Waterfall model for mobile medical application development that ensures conformance with regulatory requirements and responds to unique market development needs, thus orienting the organization towards commercial success.

CHAPTER 1

INTRODUCTION

Background

Apple's 2007 introduction of the iPhone sparked a revolution that demanded that competitors enter the mobile computing market (Banga & Weinhold, 2014). In just five years, mobile devices grew to represent the primary computing device used throughout the world (Ciaramitaro, 2011). Hence, it is not surprising that the healthcare industry has taken note of this technology shift, understanding that it has significant implications for market competitiveness, quality of care, and organizational efficiency (Iverson & Eiman, 2013). Specifically, mobile devices and mobile applications have risen to occupy an important place in the healthcare industry. As of 2013, the Apple App Place had more than 24,000 applications classified as medical applications (McGrath & Scannaill, 2013). Comprising nearly 3% of the Apple App Place inventory during the same year, mobile medical applications clearly represent a small yet significant portion of the mobile application market (Perez, 2013).

Moreover, these mobile medical applications will likely play a significant role in enhancing patient care and mitigating large-scale problems that the healthcare industry faces (Kay, 2011; Newell, 2012; Rowe, 2012). To illustrate, through facilitating virtual visits and virtual patient monitoring, these applications will help overcome the lack of access to healthcare that many patients in rural areas and developing countries face (Newell, 2012). Moreover, mobile medical applications will help promote patient engagement by enabling patients to access information and track vital signs such as blood

pressure and glucose levels. This will make patients true players in terms of health promotion, increase levels of patient safety, and lead to enhanced levels of wellbeing (Rowe, 2012). Likewise, mobile medical applications will automate tasks such as when doctors need to send prescriptions to pharmacies, or when patients need to order their own prescription refills (Kay, 2011). Such advances will lead to a reduction of resources needed to tend to a specific patient, better situating health professionals to handle the increased caseloads that will inevitably accompany the population increases occurring throughout the world. Moreover, the development of mobile applications that work with peripheral devices to transform a mobile device into a medical device such as a blood pressure monitor mean that more patients in developing countries will have access to vital medical devices (Kay, 2011). Continuing, more innovative devices stand to save the medical industry significant amounts of money. For example, the Vitality Glowcap© replaces a standard medicine bottle cap (Comstock, 2014). It works with a patient's smartphone to monitor whether a patient takes his medication. Then it notifies a health professional or family member of associated problems. Innovations such as these will help alleviate the nearly $300 billion annual costs associated with patients not taking medication correctly. Hence, it is clear that mobile medical applications stand to transform the healthcare industry.

Despite the promises that mobile medical applications offer for the medical industry, they give rise to a unique set of challenges. Specifically, there is a pervasive lack of quality assurance (QA) efforts throughout the mobile medical application industry. The implications of this are dire. One only needs to look at the disastrous

rollout of HealthCare.gov to understand the consequences of failing to implement a quality system (QS), for experts throughout the technology industry attribute the website's initial failure and ongoing difficulties to the absence of QA activities during the design process (Weintraub, 2013). While this failure did not directly affect the health of any individual patient, such a failure could have deadly consequences when designing a mobile medical application that either operates in conjunction with a medical device or turns a mobile platform into a medical device.

Regardless of how important QA efforts might be, meeting these needs for the mobile medical application market has proven to be difficult. Compounding this difficulty is the need of mobile medical application manufacturers to pursue commercial success while also going to great lengths to ensure patient safety and conformance to regulatory requirements. Specifically, with any mobile application, market demands dictate a short product development lifecycle of a few weeks to a month (Nicol, 2013). Concurrently, companies face demands for exceptionally high quality levels. This is partially attributable to user ratings of applications, which play a significant role in determining application visibility in the marketplaces (Nicol, 2013; Rowles, 2013). Specifically, positive reviews ensure that an application is more likely to appear in results for searches performed in the marketplaces. Hammond (2013) points out that several one star reviews can quickly snowball, spelling a quick demise for an application. Thus, organizations must consistently strive to develop applications that secure high user ratings. In order to orient their company towards commercial success, software

responsiveness to changing product requirements, early and ongoing product testing, shorter time-to-market, and prototyping (Lin & Fan, 2009; McCormick, 2012; Scharff, 2011). As a result, application of Agile practices, tools, and principles promises to facilitate commercial success (Lin & Fan, 2009; McCormick, 2012).

Augmenting the Waterfall model through the application of Agile Practices holds several promises for mobile medical application developers that will help ensure commercial success and conformance to safety and regulatory requirements. First, such an approach yields predictable application delivery with minimal defects and higher customer satisfaction levels (Hewlett-Packard Development Company, 2012). Second, the approach provides the opportunity to utilize automated testing both early in the product lifecycle and continually during the product lifecycle. Third, the approach helps shorten product development lifecycles. Finally, the approach maintains the formal documentation, predefined lifecycle, and clearly defined processes so important to ensuring regulatory compliance. Because of these promises, the author discusses the true potential that this hybrid Waterfall-Agile model holds for mobile medical application developers in this thesis.

Statement of the Problem

Though the mobile medical application market has only existed for a few years, it is an extremely volatile and competitive market due to a number of factors (Ciramitaro, 2011). First, mobile medical application development stands as very different from traditional medical software development, for mobile medical applications are characterized by immediacy and engagement, touch multiple business processes and

developers must work in such a manner to meet these strenuous market demands for mobile medical applications.

Specifically, manufacturers must contend with Food and Drug Administration (FDA) regulatory requirements stipulating that all applications that operate in conjunction with a medical device or transform a mobile platform into a medical device must pass FDA review procedures (U.S. Food and Drug Administration [FDA], 2013). Moreover, the FDA directs manufacturers to adhere to certain design controls and choose a QS, which is a method for designing, producing, and distributing a product that is safe, effective, and cost-efficient. Lastly, the FDA requires manufacturers to rely upon current good manufacturing practices (CGMP's) when choosing a QS (Krouse, 2012).

Hence, mobile medical application developers face a dilemma in which they must simultaneously work to secure commercial success and ensure conformance to FDA and other safety requirements. This means implementing shorter product development lifecycles, fulfilling escalating demands for quality, and constructing items far more complicated than previous types of software (Nicol, 2013). These competing demands render the long-time standard QS for software development, the Waterfall model, insufficient for guiding the design of mobile medical applications (Rasmussen, Hughes, Jenks, & Skach, 2009; Scharff, 2011). Specifically, though the Waterfall model has a long history of ensuring patient safety and conformance to regulatory requirements, it is not well suited to the development of mobile medical applications which demand shorter product development lifecycles, a greater responsiveness to changing product requirements, and unusually high customer demands for quality (Nicol, 2013).

Consequently, to proceed in the mobile application market, companies must objectively look at their organizational needs and processes to create a QS that simultaneously secures commercial success and ensures conformance to safety and regulatory requirements.

Despite these difficulties, it seems that an organization can secure commercial success while also ensuring conformance to regulatory requirements by implementing a modified QS that combines Agile practices, tools, and principles with the Waterfall method. To explain, manufacturers have long used the Waterfall model, a planned software development approach consisting of between five and seven phases, for traditional medical software design and development. The method proceeds much like a waterfall, meaning that once a phase is completed, the project team moves forward with no plans to revisit the phase. As such, there is significant time invested in each phase, with strong trust invested in the belief that a significant initial time investment prevents problems before they occur. Researchers praise the method for its notable formal documentation procedures, formal structured product lifecycle, and well-defined processes that leave room for few surprises (Munassar & Govardhan, 2010; Taneja, Sarpal, & Arora, 2013). More pointedly, use of the Waterfall model will help ensure patient safety and adherence to strict regulatory requirements.

In contrast, the Agile approach encompasses a set of practices and beliefs that serve as the foundation for several different approaches to software development (Dyba & Dingsoyr, 2009). Though the Agile approach is not without its shortcomings, it has several promising outcomes. Specifically, incorporation of Agile practices facilitates

partner services, rely upon erratic and unreliable cellular networks, and require quick streamlined development with as few steps as possible (Banga & Weinhold, 2014; David & Murman, 2014). Moreover, several researchers suggest that the look and feel of mobile applications is just as, if not more important to determining product success than it is with traditional software (Ickin et al., 2012; Wasserman, 2010). Simultaneously, there are low barriers to market entry, meaning that there is a constant influx of competitors looking to create a better application (Nicol, 2013). Finally, mobile medical application development requires the input of various stakeholders.

Additionally, mobile medical application manufacturers must also contend with regulatory requirements. The FDA's Quality System Regulation (QS Regulation) mandates the use of CGMP's for the design, development, and manufacturing of mobile medical applications (FDA, 2014a). The Waterfall model was a CGMP that more than met traditional software developers' QA requirements. It fulfilled the desire to provide safe, effective, cost-efficient, and competitive medical applications. Moreover, it ensured compliance with FDA requirements (Lin & Fan, 2009). Unfortunately, using the Waterfall model for mobile medical application development has undermined the ability of many companies to conduct business in a cost-efficient and competitive manner. The Waterfall model compromises quality and efficiency, for it employs testing only at the end of the design and development process (Rasmussen et al., 2009). Furthermore, the lengthier product development lifecycles employed when using the method inhibit the ability of mobile medical application manufacturers to meet the market expectation for constant deployment of new application versions (Lin & Fan, 2009). Finally, the method

is extremely resistant to changing product requirements. As such, competitors are quickly outpacing businesses that continue to use the Waterfall model.

Hence, it is clear that a significant problem facing today's mobile medical application developers is the lack of a QS that ensures conformance to regulatory requirements while also orienting design processes so as to ensure high degrees of quality and efficiency.

Purpose of the Study

The author's primary purpose in this study was to develop a QS applicable to the development process of mobile medical applications. This QS needed to build upon CGMP's and preserve several aspects of the Waterfall model so as to comply with the FDA's QS Regulation, as well as incorporate successes from Agile methodologies so that the organization would be oriented towards market competitiveness and operational efficiency. Through developing a QS for the design and development process for mobile medical applications that draws upon the successes of the Waterfall model while also applying aspects of the Agile methodologies, an organization would enjoy the following benefits:

- a design process that responds to changing product requirements, involves minimal process costs, ensures a satisfying customer experience, and entails less rework near the project end-date;
- a design and development process that reveals product limitations early in the design process and integrates communication to all stakeholders such that they understand how the product works;
- an unparalleled level of quality while delivering just enough functions to fulfill customer requirements;

- an enhanced risk management process that results in greater risk predictability and prioritization, as well as ongoing parallel risk assessment; and
- an enhanced level of productivity (Nicol, 2013).

In order to accomplish this goal, the author completed the following tasks:

1. identified the specific quality needs of the present mobile medical application market;
2. identified aspects of the Waterfall model which appropriately fulfill the interests of mobile medical application developers;
3. identified aspects of the Agile Methodology that have been successful at enhancing the manufacturing process of mobile medical applications;
4. used knowledge of CGMP's and the findings of step one, two, and three to develop a QS to be used for mobile medical application manufacturing;
5. illustrated an application of the newly developed QS to the design and development of an existing mobile medical application; and
6. rationalized use of the newly developed QS for the design and development of mobile medical applications based upon the enhanced results that it offers.

The author discusses the findings concerning these tasks in this thesis. Once created, the new QS functioned to address the current needs of the mobile medical application market, facilitate adherence to regulatory requirements, and orient a given company towards competitiveness.

The author chose to preserve aspects of the Waterfall model when creating a new QS for mobile medical application development due to Waterfall's formal product development lifecycle, defined processes, and thorough documentation procedures (Lin & Fan, 2009). Concurrently, the author chose to augment Waterfall with Agile methodologies that better fulfill market requirements for mobile medical applications (e.g. Lin & Fan, 2009; Rasmussen et al., 2009; Scharff, 2011). For example, Lin and Fan

(2009) explain that Agile's ability to respond to changing customer requirements makes it well suited to the design of FDA regulated mobile medical applications. Likewise, Rasmussen et al. (2009) and Scharff (2011) attribute Agile's superior ability to respond to market demands to the continued availability of working software, which in turn facilitates ongoing testing and shorter development cycles.

Theoretical Bases and Organization

The author found the theoretical basis for this study in both the Waterfall and Agile methodologies. The Waterfall model was of particular interest in this study because it has long been the standard QS used for traditional medical software design and development (Lin & Fan, 2009; McCormick, 2012). McCormick (2012) explains that the Waterfall model utilizes six phases in which development moves from one phase to another in the same way as a waterfall; that is, once a phase is completed, the project team does not return to it. The assumption underlying this approach to software development is that considerable time spent at each stage ensuring fulfillment of all requirements serves as insurance that the project team will not later have to spend time fixing mistakes. Moreover, the Waterfall model has the advantage of employing a formal and well-defined system that includes predefined processes. Finally, use of the Waterfall model is conducive to ensuring conformance to regulatory requirements due to its extensive formal documentation procedures, formal and structured product development lifecycle, and defined processes (Lin & Fan, 2009).

The Agile Methodology is in fact an umbrella term that refers to various different QS's that adhere to a common set of principles. When developing a QS for this thesis,

the researcher accounted for several Agile principles, the most important of which were collecting comprehensive application requirements only at the time that tangible software development begins, maintaining an iterative work style that stresses the need to consistently deliver working software and continually capture new product requirements, retaining a responsiveness to change as opposed to sticking to a preconceived plan, and conducting ongoing quality testing from the beginning of and throughout the product lifecycle (Agile Alliance, 2001; Blankenship, Bussa, & Millett, 2011). These principles have the potential to enhance the Waterfall model so as to create a QS suited to the needs of mobile medical application development (e.g. Hewlett-Packard Development Company, 2012; Gartner, 2009; Rasmussen et al., 2009; Scharff, 2011).

Moreover, this study relied upon several QS's that exemplify application of Agile methodologies, the most notable of which were Scrum and Extreme Programming (XP). Scrum dates back to Takeuchi and Nonaka (1986), who developed a team-based approach to product development that stressed the need for the project team to have autonomous decision-making power. Later, Sutherland and Schwaber (1997) modified Scrum by organizing and managing the simultaneous flow of different types of work that are all associated with one software project, thus accelerating the product development lifecycle. Scrum employs a series of iterative *sprints*, which are two to four week product development cycles that typically focus upon one or two features and produce some sort of working, distributable software (Blankenship et al., 2011). Moreover, the developer uses customer input to assign priority status to certain design requirements.

Finally, Scrum requires that a product owner be present for the entire design process, placing special importance on the individual customer's experience with an application.

While Scrum largely focuses on project management, XP focuses upon developmental details, stressing the need for short development lifecycles, a lean process, and customer satisfaction (Blankenship et al., 2011). As such, XP and Scrum are largely complementary and are often used together to ensure customer satisfaction. Moreover, XP addresses the software engineering process, stressing a need for the following activities when designing and developing software: coding of software, testing of software, listening to customer requirements, and creating a design structure that organizes the system. Finally, like Scrum, XP organizes these activities on a small iteration-by-iteration basis, which results in the delivery of useable software on a continual basis.

Hence, a strong foundation in the Waterfall model, the Agile principles, and any observed successful examples of QS's that operate under Agile methodologies such as Scrum and XP made up the theoretical basis of this study. These theories served to guide the development of a QS applicable to the design process for mobile medical applications.

Limitations of the Study

The specific QA needs of an organization should guide the design, development, and implementation of a QS based upon both the Waterfall and Agile methodologies. In this case, the constructed QS fulfills mobile medical application design and development needs. While this QS may be applicable to the design process for mobile applications

that address health concerns, one should ensure that it fulfills the organization's QA needs. As such, one should be cautious when generalizing the findings of this study to another situation.

At the same time, the mobile medical application environment is rapidly changing. Due to this reality, one must be rather prudent when applying the findings of this study to other situations. As such, companies must be constantly careful to understand the changing requirements of all stakeholders, modifying the QS as necessary.

Finally, Agile methodologies encompass a large number of strategies, for which it would have been impossible for this study to account for every single one. As such, QA professionals ought to conduct further research to determine how the full spectrum of QA tools can best fulfill the needs of mobile medical application development.

Definition of Terms

Agile Manifesto: written in February of 2001, the Agile Manifesto represents the consensus of seventeen experts in software programming who together pursued a better method of software design. The Agile Manifesto contains a loose set of core beliefs and principles developed for the purpose of guiding software development (Agile Alliance, 2013).

Agile Methodologies: these are the frameworks and processes developed in response to the rigid processes of planned software development methods (McCormick, 2012). Software developers designed the Agile methodologies based upon the principles set forth in the *Agile Manifesto.*

Class-Responsibility-Collaborator (CRC) Cards: individual cards listing a class or part of the completed mobile application, the responsibilities of that part, and the parts with which the given part interacts (Agile Alliance, 2013). With the cards, the project team can work with stakeholders who are not software developers to model the completed system and identify any problems.

Context-Driven Testing: this is testing done in such a manner so as to model the conditions and environment in which the customers will likely use the application (Agile Alliance, 2013). This type of testing is particularly helpful for identifying problems that might occur under sub-optimal operating conditions.

Continuous Deployment: this practice focuses on decreasing the amount of time between the writing of code for the application and when the application is released to the public (Agile Alliance, 2013). Continuous deployment results in the team creating working applications on a more frequent basis, which in turn translates into value for the company.

Critical Path: this is the minimum amount of time needed to complete all interdependent tasks for a given project (Agile Alliance, 2013).

Current Good Manufacturing Practices (CGMP's): acceptable QA practices for medical device design and development as determined by the FDA (FDA, 2014a).

Daily Stand-Up: this is a short fifteen-minute meeting that occurs each day of project development at a given time (Agile Alliance, 2013). This meeting stands as an opportunity for project members to update the team regarding any completed tasks and any problems or difficulties.

Extreme Programming (XP): this is an Agile technique that focuses upon the details of development, stressing a need for short development cycles, a lean process, and customer satisfaction (Blankenship et al., 2011). XP employs an iteration-by iteration work method that stresses the need for short product development cycles that focus upon single features and the regular distribution of product versions.

Incremental and Iterative Development: this is the practice of developing a given application or piece of software in parts (Agile Alliance, 2013). That is, the project team develops the completed project in increments and works on those parts in an iterative nature. The project team works on one or several parts of the completed project at a time, completes that part, and then begins again, developing more of the completed application using the product of the previous increment.

Milestone Retrospective: at the end of the project, the project team performs a milestone retrospective, which is a one to three day meeting in which the project team analyzes the main events of project development (Agile Alliance, 2013). For the project team, this serves as an opportunity to identify team strengths and weaknesses, and then identify plans for improvement on weaknesses.

Mobile Application: these are software programs that a person can download and access using a smartphone or other mobile device such as a tablet computer or mp3 player (Banga & Weinhold, 2014).

Mobile health (M-health): these are mobile computing applications and the associated medical sensors and communications technologies used in healthcare (Istepanian, Hovanov, & Zhang, 2004). It is a far-reaching industry that transcends multiple

disciplines and includes a broad range of technologies. A number of devices, applications, and communications technologies are currently emerging that combine to create the m-health system.

Mobile medical application: for the purpose of this piece, the author defines a mobile medical application using the FDA's definition of a regulated mobile medical application as a mobile application used in conjunction with a normally regulated medical device, or a mobile application that turns a mobile platform into a normally regulated medical device (2013).

Pair Programming: this is the practice of having two programmers share the same workstation and tasks during project development. One programmer works on coding while the other watches, checks, and confirms the original programmer's work.

Product Backlog: this is a list of all requirements that the project teams must include in the final product (Agile Alliance, 2013). Generally, the project team orders items on the product backlog according to priority. The project team works off the product backlog throughout the project.

Product Owner: this is the practice of keeping a customer representative on-site and available for questions or input throughout the development process (McCormick, 2012). This ensures that customer requirements are clear and constantly updated.

Project Vision: this is a written piece that serves as a central and shared goal for the project team during design and development processes (Bassil, 2012). It represents a shared understanding of where the project is going and what it will look like when the team gets there.

Quality Assurance (QA): this term refers to all planned or systematic actions and the associated tools that a project team uses to ensure with an adequate level of confidence that a process, product, or service will satisfy customer expectations and any relevant regulatory requirements (Omdahl, 1997).

Quality System (QS): this is the management structure, tasks, processes, procedures, activities, proficiencies and resources that are used to ensure that production and service satisfy customer requirements or other needs (Omdahl, 1997). A QS ought to utilize CGMP's.

Scrum: this is an iterative approach to software development that employs a set of Agile techniques that work to coordinate project management (Blankenship et al., 2011).

Requirements Prioritization: this is a process whereby the project team determines the order in which it will work on product features. Also, this process results in the division of product features into different releases of the product.

Software Quality: a software product's entirety of features that determine the product's ability to satisfy given needs or meet specifications (Omdahl, 1997).

Sprint Backlog: similar to the product backlog, this is a list of all requirements designated for completion during a given sprint or iteration (Agile Alliance, 2013).

Sprint Retrospective: at the end of each sprint, the project team performs a sprint retrospective, which is a meeting lasting a few hours in which the project team analyzes the events of the iteration (Agile Alliance, 2013). For the project team, this serves as an opportunity to identify team strengths and weaknesses, and then identify plans for improvement on weaknesses.

Sprint Review: at the end of each sprint, the project team conducts a sprint review, which is an opportunity for the team members to present the user stories completed during the sprint. In addition, the project team presents an informal demo of the product, thus providing an opportunity for customers to provide valuable feedback on the product.

Test-Driven Development (TDD): this describes a development approach in which software developers write tests for a given feature prior to beginning development on the feature (Agile Alliance, 2013). TDD generally results in a significant reduction in bugs.

User Stories: this is a manner of defining design requirements with one or two sentences using everyday language (Agile Alliance, 2013). The user story defines the why, who, what, and how of a user requirement.

Waterfall Model: it addresses the development cycle by breaking it down into phases, each with a rigid beginning and end (Blankenship et al., 2011). The project team must fully complete each phase before the next one begins. Upon stage completion, the project team does not revisit the feature for changes or updates. It is a plan-driven method, and it was consistently used not because it was the best suited to software development, but rather, because it was one of only a few choices that were all inferior to it.

CHAPTER 2

REVIEW OF LITERATURE

This research started with an assessment of the needs specific to mobile medical application design and development, which first entailed a review of the FDA publication, *Mobile medical applications: Guidance for industry and Food and Drug Administration staff,* and the QS Regulation, as well as associated scholarly literature and industry publications. This research continued with an examination of the needs unique to the mobile medical application market. Next, this research moved forward with a review of the Waterfall Model—the industry standard QS for both traditional medical software and mobile medical application development—that identifies where the methodology has succeeded and failed at meeting the needs of mobile medical application development. Finally, this research concluded with an examination of the Agile methodologies because they offer a promising means for promoting competitive and efficient mobile medical application development.

Food and Drug Administration Regulatory Documents

Where mobile medical applications are concerned, the FDA has published, *Mobile medical applications: Guidance for industry and Food and Drug Administration staff,* and the QS Regulation. While these are in part complimentary, each has its own set of implications for mobile medical application regulation.

The Quality System Regulation

The QS Regulation establishes minimum requirements for the design, development, and manufacturing of medical devices (FDA, 2014a). Due to the extensive number and types of medical devices, the QS Regulation establishes a flexible framework that requires companies to implement a QS when designing and developing certain medical devices. The QS Regulation directs manufacturers to rely upon current good manufacturing practices (CGMP's) and reasonable judgment when determining how to implement different aspects of the QS Regulation framework to ensure that a device consistently meets relevant requirements and specifications. The FDA describes CGMP's as the most suitable, current, and sophisticated manufacturing practices for the specific device. Hence, this reveals that there is a great deal of flexibility built into the QS Regulation.

For the purposes of regulation, the QS Regulation classifies medical devices into the following three classes:

- Class I Devices: These devices do not present a risk to humans, do not help sustain life, and are not instrumental in preventing harm to human health (FDA, 2014b). These devices are only subject to general controls.

- Class II Devices: These devices present a moderate risk to humans and must meet higher standards than Class I devices (FDA, 2014b). As such, design and development processes for such devices must ensure to a high level of certainty that they will perform as expected without presenting a risk of injury or harm to the user. These devices are subject to general controls, as well as some special controls such as labeling requirements and adherence to minimum performance specifications.

- Class III Devices: These devices present a high risk to humans, and manufacturers must adhere to general and special controls, as well as obtain pre-market approval to ensure the device's effectiveness and safety (FDA, 2014b). These devices either help to sustain life or are instrumental in preventing harm to human life.

Mobile Medical Application Regulations

Until the 2011 draft publication and the 2013 final publication of, *Mobile medical applications: Guidance for industry and Food and Drug Administration staff,* the FDA had provided limited guidance for mobile medical application design and development. In the publication, the FDA explains that it plans to regulate only mobile medical applications that meet one of the following criteria:

- mobile applications that enable the platform to connect to a normally regulated medical device for the purpose of controlling it or analyzing, displaying, storing, or communicating patient-specific data;
- mobile applications that transform a mobile device into a regulated medical device; or
- mobile applications that provide patient-specific diagnosis and/or treatment recommendations (FDA, 2013).

Additionally, the FDA specifies that a mobile medical application must meet regulatory requirements according to its classification as a class I, II, or III medical device.

Quality System Requirements

Since mobile medical applications falling into the regulatory scope of the FDA are subject to the same regulations that exist for medical devices, it is important to understand the QS requirements and the associated design controls. To start, organizational management must institute a thoroughly documented software design and development process that dictates how the creation of and revision of software occurs (Rasmussen et al., 2009). This results in more predictable and consistent outcomes. In order to provide more direction, the QS Regulation dictates certain design controls to which organizations ought to adhere.

Design and Development Planning. Design and development planning requires medical device manufacturers to keep plans describing to the best degree possible all activities related to the design and development process, as well as responsibility for implementing each activity (FDA, 2014a). This plan must serve as a framework for ensuring that all aspects of the design and development process conform to QS Regulation requirements. It is a living document, and the project team updates it as the design process proceeds. Moreover, the FDA requires submission of this document in order for the mobile medical application to gain premarket approval (FDA, 2014a; Kinsel, 2012).

Design Input. These are the physical or performance characteristics of a device that are used as the basis for all successive design and development activities, as well as for design validation (FDA, 2014a). In order to create the design input, the QS must include procedures for gathering design requirements that address the intended use of the device, fulfill the needs of the patient and any other users, and ensure adherence to any regulatory requirements. Finally, when considering design inputs, it is important to have a process that addresses any conflicting or incomplete requirements.

Design Output. The design outputs for a medical device are the outcomes of each phase of the design and development process, as well as the outputs of the entire process (FDA, 2014a; Kinsel, 2012). Design outputs start very broad and grow more specific as the design process concludes. Any design output documents must define the design output in manners that permit assessment of whether the observable design outputs conform to design inputs.

Design Review. Design review is an ongoing activity accompanied by documentation that occurs throughout each of the design and development phases (FDA, 2014a; Kinsel, 2012). Each design review assesses the capability of the design to fulfill design input requirements. Moreover, this activity helps identify aspects of the design which may cause a manufacturing problem or do not meet the needs of the patient or user.

Design Verification. Design verification is the process of conducting activities and creating records to confirm that the actual theoretical design outputs as specified on paper meet design input requirements (FDA, 2014a; Kinsel, 2012). Examples of verification activities are testing, inspection, and analysis. Like all other design controls, the design verification process must adhere to written procedures.

Design Validation. Similar to design verification, design validation is the process of conducting activities and creating records to establish objective evidence that the actual device or mobile medical application conforms to design inputs (FDA, 2014a; Kinsel, 2012). This usually requires the building of a prototype of the finished product. Examples of validation activities are testing, inspection, and analysis. Like all other design controls, the design validation process must adhere to written procedures.

Design Transfer. Design transfer is the process of translating design specifications into production specifications (FDA, 2014a; Kinsel, 2012). Like other design controls, design transfer activities must follow written procedures. Design transfer consists of creating any resources that help direct production such as paper or

digital specifications for the final product, test and inspection processes and programs, and programming files.

Design Change. Design change is the process of identifying design deficiencies, and then formulating, reviewing, and approving design changes (FDA, 2014a; Kinsel, 2012; Lincoln, 2010). This phase of design and development consists of two major activities—document control and change control. Document control involves approving, reviewing, updating, and organizing design documents, as well as tracking of design revisions and preventing the use of obsolete documents. Similarly, change control involves approving, reviewing, and organizing changes in design requirements. Additionally, change control involves the tracking and dissemination of any changes to the design plan resulting from changes in design requirements, as well as verification and validation activities to ensure that the final design incorporates changes in design requirements.

Design History File. The design history file (DHF) is a compilation of all documents that have been part of the design and development process (FDA, 2014a; Lincoln, 2010). The DHF shows that design activities occurred in accordance with the QS Regulation and procedures set forth during design and development planning.

Analysis of Regulatory Requirements

Academics and industry professionals have voiced concern that FDA regulations will stifle innovation in the mobile medical application industry and cause development costs to increase (Barton, 2012; Danzis & Pruitt, 2013; Yetisen et al., 2014). Similarly, though the market requires development at a speed rarely paralleled in other industries,

the FDA approval process requires a significant time investment. With the rate of change in the mobile application market so fast, there is concern that an application will be unable to run efficiently by the time of release. Moreover, new regulatory requirements make it difficult for companies without a history of working with FDA pre-market approval to compete in the market (Yetisen et al., 2014). Hence, the FDA regulations may increase barriers to market-entry to such a degree that lack of potential competitors slows market innovation and impinges upon the ability of manufacturers to innovate (Holzer & Ondrus, 2011).

These concerns with the regulatory process are formidable. Concurrently, there are notable themes woven throughout the QS requirements. This analysis indicates that formal written procedures and record creation and maintenance is paramount to ensuring adherence to the QS Regulation. Moreover, it is important to understand there is a great degree of flexibility built into the regulatory requirements because of instructions that merely require that manufacturers rely upon CGMP's. Hence, it is clear that mobile medical application developers must have QS options that promote navigation through the regulatory maze while also orienting companies towards market competitiveness.

Mobile Medical Application Market

Not surprisingly, mobile devices constitute the primary computing device used throughout the world. As of 2010, 68% of the world's population had access to a mobile computing device (Ciaramitaro, 2011). Businesses have acknowledged this, and several industries have set their focus upon the mobile application, significantly changing the structure of industry as most people know it (Iverson & Eiman, 2013). With growing

interest in the mobile application, stakeholders have noticed that the mobile application market is unique and presents with significant challenges for QA professionals. As is shown below, this uniqueness and the associated QA challenges only grow as one moves into the mobile medical application environment with its regulatory hurdles.

Multiple Platforms and Devices

As the mobile market has grown, so too has the number of mobile platforms (operating systems) and devices. As such, mobile application makers invest significant resources in ensuring that applications can run on at least the main mobile platforms—Android, iOS, and Windows Phone (e.g. Holder, 2013; Joorabchi, Mesbah, & Kruchten, 2013; Wasserman, 2010). In addition, manufacturers who compete in Android's Google Play store or the Windows Marketplace must outfit their applications to run on an endless number of devices with varying specifications (Joorabchi et al., 2013). This differentiates mobile applications from desktop software, for publishers of the latter only need to address the differences between the Apple and Windows environments (Wasserman, 2010). This results in a significant increase in the resources needed for mobile application design and development. The need to respond to mobile operating system updates, which occur as often as once a month, further taxes organizational resources. Hence, this emerges as a significant challenge in the mobile medical application world, for there remains some question as to whether designers of class III medical applications would need to conduct a separate FDA approval process for each version of an application.

Device Network Speed and Security

Several academic and industry publications identify cellular network bandwidth capability and reliability as a concern when designing mobile applications (e.g. Dehlinger & Dixon, 2011; Holzer & Ondrus, 2011; Wasserman & 2010). Whereas desktop software generally uses information stored directly on the device, mobile applications use a device's cellular network connection to access services and data on the internet (Wasserman, 2010). A substantial amount of literature and user experience indicates that there is large variability in cellular network connection speeds and types (Ravindranath et al., 2013; Gember, 2012). This has significant implications for mobile application design and development. Some mobile applications may behave differently depending upon the network speed available at a given time (Wasserman, 2010). The possibility of variable network capability resulting in delayed, faulty, or lack of performance has real implications when considering mobile medical applications that affect the immediate wellbeing of a patient. Likewise, variability in network type prompts questions regarding the security of patient information (Unhelkar & Murugesan, 2010). In a society that places a strong emphasis on patient privacy, possible security issues resulting from different types of networks emerge as a great concern. The QS developed in this piece must strive to address this issue.

Unparalleled Customer Quality Requirements

Several researchers and industry professionals describe the mobile medical application market as having incredibly high customer demands for quality (e.g. Andjelkovic & Imaizumi, 2012; Holzer & Ondrus, 2011; Nicol, 2013). This is

attributable to low barriers to market-entry and application ratings that quickly foretell the fate of any given mobile application. Andjelkovic and Imaizumi (2012) explain that mobile application design and development occurs in a large variety of places. Anyone from members of a programming team at a large organization to a freelance programmer working from a home office in a developing nation can develop a mobile medical application. As such, the mobile medical application industry emerges as a fertile starting ground for entrepreneurs who can find a way to mitigate the costs associated with regulation. The resultant reality is that a high level of competition where it is mandatory to both meet and exceed customer demands for quality typically characterizes the mobile medical application market.

Moreover, the rating systems used throughout the application marketplaces have proved instrumental in determining the success or quick demise of a mobile application (Nicol, 2013). Even for applications that have met all regulatory requirements, negative reviews in one of the marketplaces can quickly spell out a negative fate. Nicol (2013) explains that this is particularly the case when QA activities have been absent from design and development activities, which often results in one or two initial bad reviews that are echoed by a few more bad reviews. With many competing mobile medical applications available due to the low barriers to market-entry, users prefer to install applications which either have positive reviews or no reviews instead of installing applications that have even a small number of poor reviews. As such, competing in this market requires that mobile medical application developers commit significant organizational resources to developing applications that function seamlessly while also

integrating attractive features and interfaces that exceed customer requirements (Andjelkovic & Imaizumi, 2012; Holzer & Ondrus, 2011; Nicol, 2013).

Short Product Development Cycles

Nicol (2013) notes that the mobile application market requires product design lifecycles far shorter than that for desktop applications. Several researchers echo this sentiment and move on to note that the typical product development lifecycle used when the Waterfall Model was applied to desktop software design is too lengthy for mobile application design and development (Corral, Sillitti, & Succi, 2013; Jeong, Lee, & Shin, 2008). This is attributable to greater competition, rapidly evolving customer demands, and constantly changing devices and platforms. Consequently, the short development cycle mandated in the mobile application market presents challenges when choosing a QS to guide design and development activities, as well as for conforming to the quality testing requirements of FDA regulated mobile medical applications (Corral et al., 2013, Jeong et al., 2008; Savanæs, Alsos, & Dahl, 2010).

Testing Difficulty

Design testing is key to fulfilling high customer expectations for quality and maintaining adherence to the FDA's QS requirements, yet the literature identifies testing as another challenge for the design and development of mobile applications (e.g. Kumar & Chauhan, 2013; Muccini, Francesco, & Esposito, 2012; Wasserman, 2010; Zahra, Khalid, & Javed, 2013). Each of the aforementioned challenges for mobile application development also emerges as a challenge during mobile application quality testing. Savanæs et al. (2010) underscored the difficulty that would arise from needing to adhere

to regulatory requirements during mobile medical application testing even before the release of FDA guidance on the matter. Effectively, normal QA testing needs enhancement to serve as an acceptable means for ensuring patient safety (Kumar & Chauhan, 2013; Savanæs et al., 2010). Both Kumar and Chauhan (2013) and Wasserman (2010) underscore the mobile application testing challenges that arise due to variation in mobile devices and platforms. Effectively, mobile medical application developers would need to test each application on a large multitude of devices and platforms in order to both adhere to the FDA's QS requirements and satisfy customer quality requirements. Likewise, large variations in network speeds and security levels mean that mobile medical application designers will need to test applications under a variety of different conditions (Kumar & Chauhan, 2013; Unhelkar & Murugesan, 2010). Finally, methods for testing desktop software are not applicable to the testing of mobile applications due to several of the aforementioned difficulties (Wasserman, 2010). Hence, the fact that mobile applications have been around for a relatively short period of time means that there is a lack of research-based testing methods available for them. Even those that show promise still emerge as troublesome due to the lack of information on them (Kumar & Chauhan, 2013).

The Waterfall Model

The Waterfall Model has long been the standard QS used for both typical software development and medical software development (Awad, 2005; Lin & Fan, 2009; McCormick, 2012). Due to this precedent, both typical mobile application and mobile medical application manufacturers have looked to it as a means for guiding design

and development (Jeong et al., 2008; Martin, Lopez de Ipina, Alzua-Sorzabal, Lamsfus, & Torres-Manzanera, 2013). Like a waterfall, when the Waterfall Model is used, work moves from one phase to another with no plans to return to previous stages. This means that a project team spends considerable time in each stage, making sure that there are no doubts about all requirements being met (McCormick, 2012). Essentially, this type of software development rests upon the assumption that "considerable time spent in initial design effort corrects bugs in advance" (McCormick, 2012, p. 3). Awad (2005) and McCormick (2012) add that there is a heavy emphasis on thorough documentation at every stage of the design and development process when using Waterfall.

The Waterfall Phases

A review of different implementations of the Waterfall Model reveals a degree of variability (Bassil, 2012; McCormick, 2012; Westfall, 2009). Specifically, there tend to be between five and seven phases in the Waterfall Model. Here, a six phase Waterfall Model is detailed, yet there are notes included which help to show how a Waterfall Model may contain five or seven phases. Due to the assumptions underlying the QS, it is important to note that the order of the phases is important.

Requirements Phase. Also known as the analysis phase, this comprises all activities that go into identifying a need for a specific product and determining the associated product requirements (Bassil, 2012; McCormick, 2012). In this phase, the project team needs to list all functional and non-functional requirements that describe the behavior and look of the hypothetical application (McCormick, 2012). For a mobile medical application, this would possibly entail gathering input from various stakeholders.

In some implementations of the Waterfall Model, both problem identification and feasibility study are included as phases prior to the requirements phase.

Design Phase. The project development team uses this phase as an opportunity to define how the product will be designed (Bassil, 2012). Software developers and designers must work both together and separately to complete design of the application algorithm, a concept design, and the graphic interface design, as well as other steps to create the architecture that will provide the foundation for the software coding that occurs in the next stage (Bassil, 2012; McCormick, 2012).

Implementation Phase. At this point, software engineers write the actual code necessary to realize the vision set out in the mobile application architecture (Bassil, 2012; McCormick, 2012). This phase concludes with the creation of an operable, testable mobile application (Bassil, 2012).

Testing Phase. The project team uses this phase to ensure that the product meets the original specifications developed in the requirements phase, as well as that it achieves its intended purpose without any defects or other problems (Bassil, 2012). Sufficient testing activities must occur to guarantee that operational and safety inputs and outputs align with the product's intended use and design specifications (Westfall, 2009). At this point, testing must also confirm that the final design implements all defined requirements. As a final point, the testing phase is sometimes broken up into separate verification and validation phases (Westfall, 2009).

Deployment Phase. This phase involves release of the actual product to the consumer (Awad, 2005; McCormick, 2012). In cases where limited deployment first

occurs, this phase may serve as an opportunity to gather feedback and return to various phases of the project development process.

Maintenance Phase. During this phase, the manufacturer provides several services, including but not limited to training, completion of enhancements to the application, and adaptations of the application to a new environment (Bassil, 2012; McCormick, 2012). Finally, it is important to note that some implementations of the Waterfall Model combine the maintenance and deployment phases.

Strengths of the Waterfall Model

Though this thesis seeks to enhance and strengthen mobile medical application design and development activities through the application of Agile practices, it is important to note that the Waterfall model has unique strengths that lend the method some notability when designing a hybrid Agile-Waterfall approach for the design and development of a mobile medical application.

Extensive Documentation. At each stage of the method, the project development team creates thorough documents that explain and describe the requirements and objectives of that phase (Munassar & Govardhan, 2010). Moreover, a thorough and written review takes place at the conclusion of each stage to ensure that the team is ready to move onto the next stage. This arduous documentation is paramount to making the Waterfall Model a great option when undertaking a project where QA and adherence to regulatory requirements is a concern (McCormick, 2012; Munassar & Govardhan, 2010). Hence, many understand this aspect of the Waterfall Model to be well suited to the needs

of mobile medical application design and development, for it fulfills the QS Regulation need for a DHF and thorough documentation throughout each phase of development.

Structured Product Lifecycle. The Waterfall Model also ensures a structured product lifecycle, which ensures conformance to the design and development planning design control (Cusumano & Smith, 1995; Munassar & Govardhan, 2010; Taneja, Sarpal, & Arora, 2013). Taneja et al. (2013) emphasize that the Waterfall Model provides clear milestones at each phase of development, an understanding of when each phase begins and ends, clear outputs at the end of each phase that serve as the inputs for the next phase, and a clear understanding from the beginning of the product lifecycle of all resources and activities necessary to see the project through to production. Moreover, Cusumano and Smith (2013) indicate that organizations tend to like the Waterfall Model because there are no surprises down the line, meaning clients have a clear understanding of the costs, time, and other resources associated with design and development activities.

Shortcomings of the Waterfall Model

While the Waterfall Model does have strengths, is also has shortcomings and limitations which make it less-than-ideal for the design of a mobile medical application.

Long Product Development Lifecycles. McCormick (2012) indicates that when the time allotted for product development is lengthy, the Waterfall Model is an appropriate QS choice; however, when the time allotment is short, it is not an appropriate choice. As mentioned earlier in this review, the mobile medical application market requires short product development lifecycles (Corral et al., 2013; Jeong et al., 2008;

Nicol, 2013). Hence, long product development cycles make the methodology a poor fit for mobile medical application design and development.

Testing. Munassar and Govardhan (2010) criticize the Waterfall Model because it does not deliver working software until late in the development lifecycle, which means that testing also does not occur until late in the lifecycle. This means that identification of large problems does not occur until late in the product lifecycle, making it costly and time consuming to go back and fix them. Sometimes this can even necessitate starting a project over from the beginning (Almani, 2012).

Unresponsive to Changing Requirements. Several researchers note that the Waterfall Model is lacking due to the model's inability to adjust to changing requirements (Cusumano & Smith, 1995; Munassar & Govardhan, 2010; Taneja et al., 2013). Specifically, Waterfall requires that all product requirements are set at the beginning of the project (Taneja et al., 2013). This may be acceptable with certain simple projects, yet in the case of the rapidly evolving mobile medical application market where customer requirements, mobile device specifications, and medical knowledge is constantly changing, a traditional Waterfall approach is not sufficient for guiding design and development activities (Nicol, 2013; Taneja et al., 2013).

Agile Methodologies

Cohen, Lindvall, and Costa (2004) emphasize that Agile symbolizes a reaction to traditional software development methodologies such as Waterfall wherein designers searched for an approach that was not linear, document intensive, and resource greedy. Dyba and Dingsoyr (2009) explain that Agile was necessary if software designers were to

deal with the computer industry where the evolution of technologies occurs too fast and requirements change too rapidly for planned software development methods to be adequate. Thus, seventeen software designers met together in early 2001 to form The Agile Alliance, and what followed was the publishing of the *Agile Manifesto,* which outlines the values, principles, and practices that comprise the Agile approach.

> The Agile Alliance begins the *Agile Manifesto* with a statement of values:
>
> We are uncovering better ways of developing software by doing it and helping others do it.
>
> Through this work we have come to value:
>
> - Individuals and interactions over processes and tools
> - Working software over comprehensive documentation
> - Customer collaboration over contract negotiation
> - Responding to change over following a plan
>
> That is, while there is value in the items on the right, we value the items on the left more (2001, p. 1).

It is important to understand that Agile is not a software development model on its own. Instead, Agile is a term used to describe a set of practices and beliefs that serve as the foundation of different approaches to software development (Dyba & Dingsoyr, 2009). Hence, the *Agile* Manifesto has spawned countless QS's, including but not limited to Extreme Programming (XP), Scrum, Adaptive Software Development, Feature-Driven Development, and Dynamic Systems Development (Erickson, Lyytinen, & Siau, 2005).

In addition to publishing a statement of values, The Agile Alliance also lists twelve principles to guide the development of Agile approaches.

- "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software" (Agile Alliance, 2001, p. 1).
- "Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage" (Agile Alliance, 2001, p. 1).
- "Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale" (Agile Alliance, 2001, p. 1).
- "Business people and developers must work together daily throughout the project" (Agile Alliance, 2001, p. 1).
- "Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done" (Agile Alliance, 2001, p. 1).
- "The most efficient and effective method of conveying information to and within a development team is face-to-face conversation" (Agile Alliance, 2001, p. 1).
- "Working software is the primary measure of progress" (Agile Alliance, 2001, p. 1).
- "Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely" (Agile Alliance, 2001, p. 1).
- "Continuous attention to technical excellence and good design enhances agility" (Agile Alliance, 2001, p. 1).

- "Simplicity--the art of maximizing the amount of work not done--is essential" (Agile Alliance, 2001, p. 1).
- "The best architectures, requirements, and designs emerge from self-organizing teams" (Agile Alliance, 2001, p. 1).
- "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly" (Agile Alliance, 2001, p. 1).

At the core of Agile is the belief that the delivery of working software via iterative work processes creates true value for a company (Cohen et al., 2004; Dingsoyr, 2012). A second emphasis in Agile is self-organizing development teams that consistently communicate through face-to-face interaction and move at a rate that best matches their creativity and productivity. Iterative work processes foster the ability to create change and adapt to changes in industry and customer requirements. Relying upon teams that value ongoing and close communication enables the project team to make decisions and act upon them immediately. Finally, Cockburn (2007) adds that Agile methodologies are light, adaptive, and ready to respond.

Agile-Based Quality Systems

As mentioned earlier, the Agile practices have given birth to several software development QS's. Two of the most notable with a high degree of relevance to mobile medical applications are XP and Scrum.

Extreme Programming. Extreme Programming (XP) focuses on software engineering, stressing a need for software coding, software testing, listening to customer requirements, and creating a design structure that organizes the system (Blankenship et

al., 2011). It employs short, iterative development cycles, incorporates a lean process, and ensures fulfillment of customer desires.

Munassar and Govardhan (2010) identify several practices as being central to XP. First, incremental planning requires the listing of product requirements on story cards. The team then determines which story cards are part of a release based upon priority and time required. Implicit to this idea are small releases of working products while on the way to the final product, as well as a simple design that only meets current customer requirements. Additionally, the team must employ a test-first approach where product tests are determined prior to developing the product. Software developers must also work in pairs, continually checking each other's work, while also being expected to refactor the code as soon as improvements are available. They then integrate the pieces into the complete system once they are complete. Finally, this approach requires that a product owner be available on-site for the entire design and development process.

Scrum. Srinivasan and Lundqvist (2009) indicate that Scrum is an approach to project management that relies upon three different roles in the design and development process. The first role, referred to as the Product Owner, has the responsibility of defining and prioritizing product requirements to create a plan for product development. The second role, the self-organizing Scrum team, does the actual work of product development. They carry it out in two to four-week long sprints that focus upon one or two product requirements. The third role, known as the Scrum Master, serves as the liaison between the Product Owner and the Scrum Team. He or she monitors the team, ensuring that it is functional and productive, and holds a daily meeting to coordinate the

activities of the Scrum team. Instead of focusing on the actual software engineering process, as is the case with XP, Scrum focuses more on project management. For this reason, most researchers and practitioners view Scrum and XP as complementary (Blankenship et al., 2011)

Shortcomings of the Agile Methodologies

Despite the popularity that Agile practices have gained since the formation of the Agile Alliance, it is important to remember that Agile does have shortcomings (Barlow et al., 2011; Dingsoyr, 2012; Lin & Fan, 2009; McCormick, 2012)

Unsuited for Large Projects. In evaluating agile, academics are quick to point out that Agile is not always the best choice when tackling larger projects (Barlow et al., 2011; McCormick, 2012). When using Agile, it is difficult to know the time and resources necessary to complete a project, which can be troublesome with large projects where costs can quickly rise into the millions. Furthermore, Agile permits constant changes in product requirements. In the case of a large project, this means that the project can easily go off track. Moreover, since there are few formal methods of communication when using Agile, it is difficult to keep everyone informed when conducting a large project. Finally, it is often difficult to apply Agile in situations where the project team is distributed over several different geographic locations, for with this there does arise several problems in regards to coordination of design efforts.

Documentation. In addition to being largely unsuited for larger projects, the very fact that Agile implies very limited methods of and emphasis on documentation is particularly troubling (Barlow et al., 2011; Lin & Fan, 2009). Specifically, this lack of

documentation means that a significant amount of information regarding a given project, sometimes referred to as tribal knowledge, resides within and among the minds of the project team members (Barlow et al., 2011). When members leave the company or move to a different project team, there is sometimes a lack of understanding regarding the logic behind certain decisions. Moreover, the lack of documentation can be detrimental to the company's ability to conduct maintenance on a given application at a later time. In addition, the lack of documentation inherent to Agile methodologies creates significant problems when pursuing FDA approval. Lin and Fan (2009) are emphatic about this belief, suggesting that compliance with FDA QS requirements requires integration of a planned software development lifecycle that includes formal methods of documentation. Similarly, Cawley, Wang, and Richardson (2010) caution that when working in an FDA regulated environment, teams should implement a hybrid QS consisting of a planned software development method that is augmented with Agile practices. First, this will ensure conformance to the FDA QS requirements by facilitating the necessary documentation. Concurrently, it provides a linear lifecycle that facilitates requirements tractability. Hence, it is clear that on its own, Agile does not include the necessary documentation of design lifecycle elements necessary to ensure conformance to the FDA QS requirements.

Promises of the Agile Methodologies

The Agile methodologies certainly present with shortcomings; however, Agile practices have yielded important benefits for software development that will be helpful for the design and development of mobile medical applications.

Respond to Changing Requirements. Relative to Waterfall and other planned approaches, Agile is more responsive to changing product requirements (e.g. Cao & Ramesh, 2008; Lin & Fan, 2009; McCormick, 2012). This is due to the iterative sprints that are paramount to Agile software development. At the beginning of each sprint, development teams capture customer requirements regarding what product features are important. Cao and Ramesh (2008) note that this results in more satisfied customers and enhanced requirement clarity and understandability. McCormick (2012) later concurred, adding that it ensures that the efforts of the team are not wasted working on product features that are no longer wanted or applicable. Scharff (2011) indicates that though these requirement gathering and prioritization activities necessitate a significant amount of time and discipline, the benefits of this process are quite noticeable during typical mobile application development. This benefit will likely transfer to mobile medical application development because frequently changing platforms and devices undoubtedly mean rapidly evolving product requirements. Finally, this consistent and thorough gathering of product requirements will ensure conformance to FDA QS requirements, with an emphasis on design input.

Ongoing Product Testing. The Agile methodologies introduce a type of testing where it is no longer seen as a last-ditch attempt to ensure quality; instead Agile emphasizes the need for all people involved in the design process to constantly test the product as it is developed (Cao & Ramesh, 2008; McCormick, 2012). Hence, in Agile approaches such as test-driven development (TDD), the project team designs tests for a specific feature before and during the actual realization of the feature in the product. The

project team also employs testing iteratively with the development of each product feature, which means that testing drives development. Moreover, developers conduct regression testing after integrating the feature into the entire project to protect against the creation of bugs. Several academics have already noted the usefulness of this testing for typical mobile application development (Jeong et al., 2008; Scharff & Verma, 2010; Wasserman, 2010). With respect to the mobile medical application market, this testing is central to delivering the quality customers expect, as well as ensuring conformance to FDA QS requirements such as design review, design verification, and design validation.

Shorter Time-To-Market. Despite the time invested in capturing changing requirements and conducting testing, academics agree that Agile offers the added benefit of permitting more efficient development practices that result in shorter time-to-market (Jeong et al., 2008; McCormick, 2012; Scharff, 2011). McCormick (2012) explains that since Agile permits adaptation to last minute changes in requirements, the common result is that products reach the market in less time than might be used when applying a planned software development method. However, it is important to be mindful of the fact that though Agile results in shorter time-to-market in most cases, this is not always the case. For instance, a planned software development approach is generally better suited for a larger project. Nonetheless, both Scharff (2011) and Jeong et al. (2008) report that application of Agile to mobile applications has resulted in shorter time-to-market. Hence, it seems hopeful that this benefit will transfer to the application of Agile to mobile medical application design.

Prototyping. Prototyping, or the frequent delivery of working software, is an integral practice when applying Agile principles to a software development model, and it is this which makes Agile such an inviting force when dealing with both traditional and mobile application development (Cao & Ramesh, 2008; McCormick, 2012; Scharff, 2011). At the end of each of the aforementioned iterative sprints, the project team delivers a working prototype of the software or application to the customer(s). In some cases, Cao & Ramesh (2008) point out that this may be a limited and very controlled release, yet with other cases, this may be a full release to the public. Regardless, the project team views it as an important opportunity to gain customer input regarding product requirements and features. Scharff (2011) points out that this practice has become so integral to the development of mobile applications that many IT professional training programs stress the need for frequent releases of a working application. Moreover, this practice provides an opportunity for ensuring conformance to design validation and design change requirements. Thus, this practice may also serve as very helpful to the design of mobile medical applications.

This literature review shows that QA professionals working in mobile medical application development must resolve to ensure conformance with regulatory requirements through exploring innovative applications of existing quality tools. This is first attributable to the fact that the mobile medical application design environment presents with a set of characteristics making it a more daunting landscape than that of any type of software development preceding it. Moreover, FDA regulations constrict the

freedom of these QA professionals, thus requiring the direction of close attention to conformance to FDA design controls.

What this means is that the QS proposed and implemented in the methods portion of this piece must draw upon the strengths of the Waterfall Model in order to ensure conformance with the regulatory requirements. At the same time, the proposed QS must integrate successful aspects of the Agile practices to ensure competitive, efficient, and appropriate design and development activities.

CHAPTER 3

METHODOLOGY

Quality System for Mobile Medical Application

The QS proposed in this thesis supports market needs for mobile medical applications by augmenting a Waterfall Approach through the application of Agile practices. As shown in Table 1, the augmentation of a Waterfall Approach through the application of Agile practices also functions to support the FDA design controls. By aligning the deliverables of each phase of the Waterfall Approach with corresponding Agile practices, it is possible to integrate the two approaches in support of the design and development of mobile medical applications. As shown in this chapter, the enhanced QS reaps proven results for adhering to and surpassing FDA regulatory requirements. Finally, the integration of the two approaches provides for an enhanced design and development process that facilitates market competitiveness.

The integration and synchronization of these approaches produces a value-added QS for design and development that consistently delivers safe and effective mobile medical applications. The combination of these approaches in practice increases organizational efficiency by delivering superior applications that satisfy FDA regulatory requirements and fulfill customer expectations. This section explains the design of the enhanced QS though identifying common deliverables, as well as the corresponding Waterfall phase, Agile practices, and FDA design controls.

Table 1

Waterfall Approach Augmented Through the Application of Agile Practices

Waterfall Phases	**Waterfall Deliverables**	**Agile Practices**	**FDA Design Controls**
Requirements	Software Development Plan Requirements Specification Functional Specification Acceptance Test Specification	Project Vision User Stories Product Owner Requirements Prioritization Product Backlog	Design Validation Design Review Design and Development Planning
Design	Preliminary Design Specifications Detailed Design Specifications Interface Design Specifications Software Test Plan Specifications	CRC Cards	Design Input Design Verification Design Review Design and Development Planning
Implementation	Source Code Listing Source Code	Daily Stand-Up Incremental Development Pair Programing Sprint Backlog Test-Driven Development Context-Driven Testing Continuous Deployment	Design Output Design Verification Design Review Design and Development Planning
Testing	Unit Test Report Sub-System Test Report System Test Report Acceptance Test Report Completed System		
Deployment	Completed System Change Request Change Request Report	Continuous Deployment Iteration Retrospective Milestone Retrospective	Design Changes Design History File Design Transfer Design Validation Design Review Design and Development Planning

Incremental and Iterative Development Process

So that the augmented Waterfall approach can bring about the benefits of Agile practices, development occurs through a modified incremental and iterative development process. That is, during the initial requirements phase, the development of a product backlog and the results of requirements prioritization processes permit an initial estimation of the number of increments or sprints needed to complete the project. Development then occurs incrementally and iteratively, with working software delivered at the end of each sprint. The practicality of this process is documented in research regarding the application of Agile to traditional software development (McHugh, McCaffery, Casey, & Pikkarainen, 2012; Rasmussen et al., 2009). The project team works through an Agile augmented Waterfall Model in an iterative manner. Figure 1 displays the iterative nature of the Agile augmented Waterfall cycle.

Working incrementally and iteratively positions the Agile augmented Waterfall model to yield enhanced design and development results. First, implementing this incremental and iterative process results in greater product release frequency, which means clients have the opportunity to provide valuable feedback while project development is still occurring (Petersen & Wohlin, 2010). Simultaneously, the augmented QS is more efficient due to a reduction in wasted time and money (Paetsch, Eberlein, & Maurer, 2012; Petersen & Wohlin, 2010). Likewise, the iterative and incremental design processes result in improvements to software quality and greater customer satisfaction.

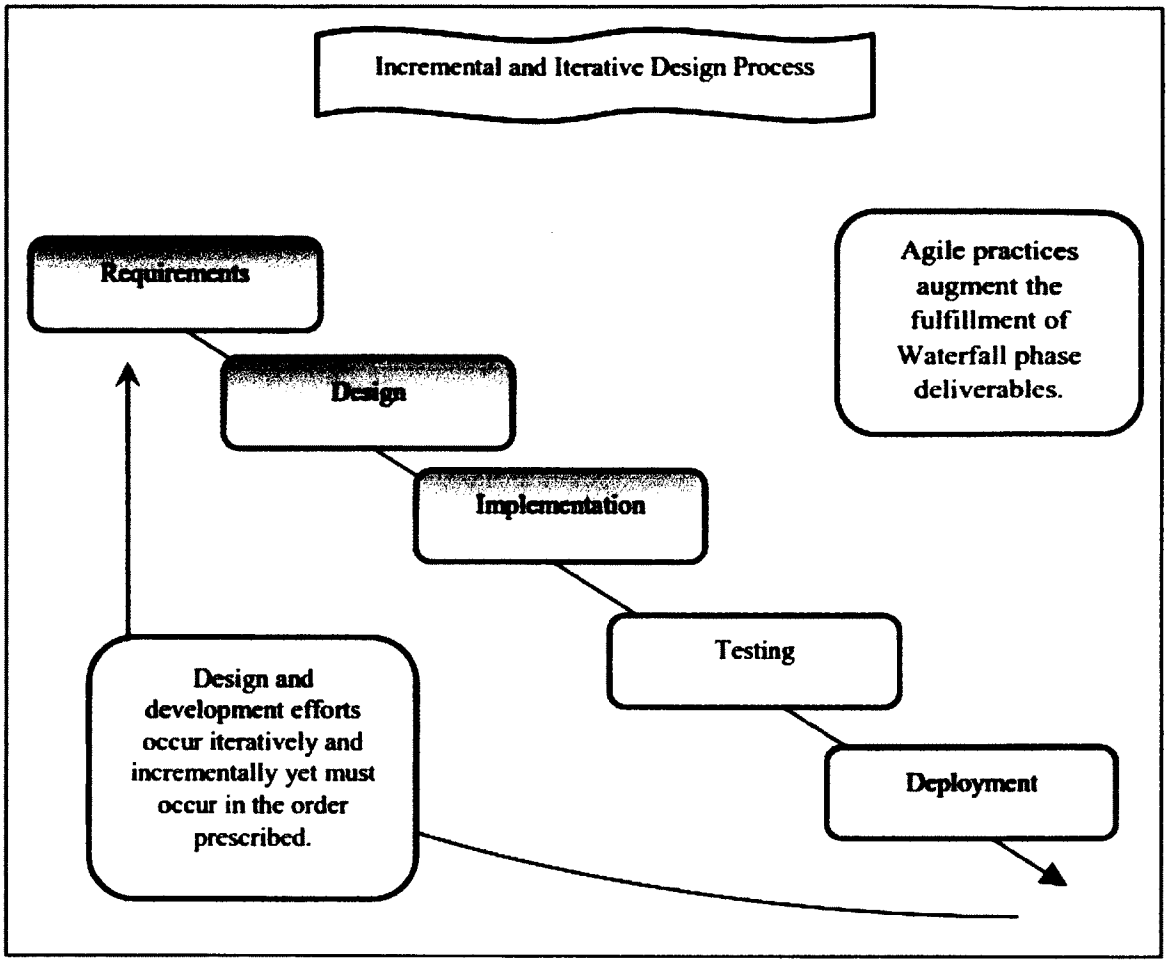


Figure 1. The incremental and iterative cycle of the augmented Waterfall approach.

However, it is important to note that this not a free-flowing iterative model like that traditionally used with Agile practices. Instead, the project team will determine the number of increments and items for completion in each increment during the first requirements phase. Once the project team makes this decision, it must undergo a formal process to change the number of increments or the items for completion in each increment. There is time to carry out such a process at the beginning of each sprint, as the project team may need to adjust the design requirements because of a change in

mobile devices or platforms. Additionally, the project team maintains the Waterfall product lifecycle within the confines of each iteration to permit conformance to FDA design controls (Lin & Fan, 2009). Finally, each iteration lasts between six and eight weeks, as any shorter amount of time does not allow time for documentation while any longer amount of time leads to a loss of project focus (Rasmussen et al., 2009).

Agile Augmented Waterfall Phases

This section describes each phase of the Agile augmented Waterfall model with an emphasis upon how Agile practices, tools, and principles facilitate realization of the Waterfall deliverables and conformance to FDA design controls.

Requirements Phase.

Figure 2 displays the relationship between Agile practices, Waterfall deliverables, and FDA design controls in the requirements phase of the augmented Waterfall QS. As is shown, the project team uses a project vision, user stories, and the input of a product owner to create a product backlog. The project team then uses requirements prioritization to enhance the product backlog. Implementation of these Agile practices ensures more accurate capturing of product requirements (Cao & Ramesh, 2008; Lin & Fan, 2009; McCormick, 2012). Moreover, the incremental and iterative nature of the augmented QS means that the QS allows for greater responsiveness to changing product requirements.

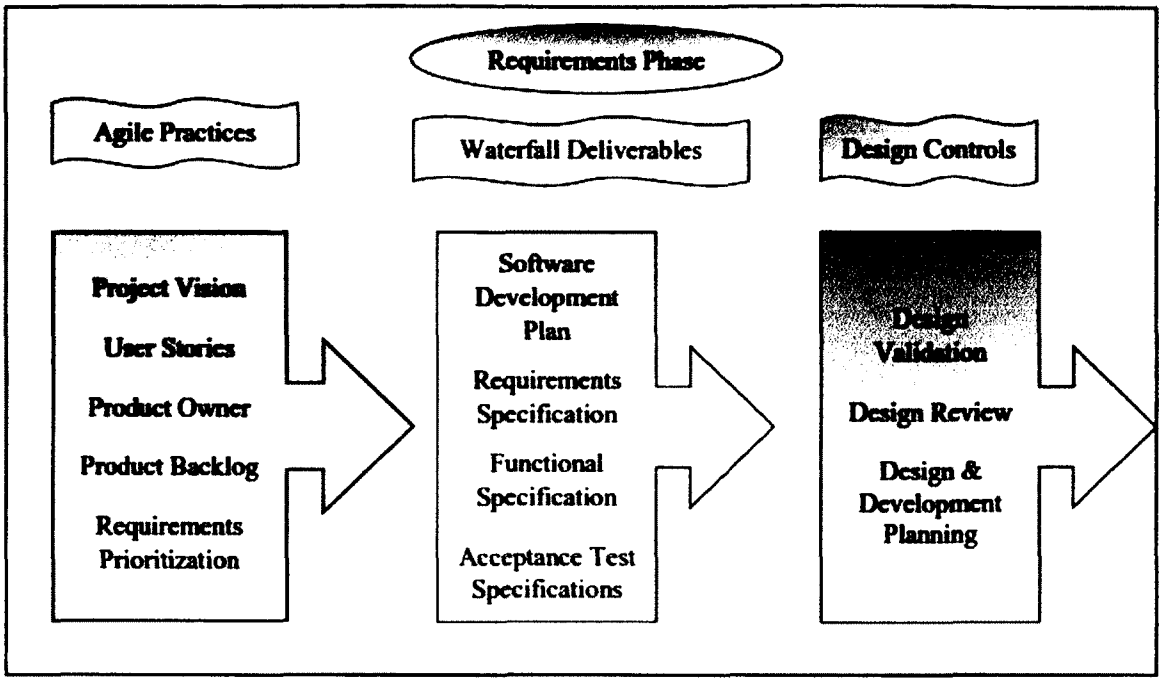


Figure 2. The requirements phase of the Agile augmented Waterfall model.

Division of the prioritized product backlog into different increments facilitates development of the software development plan document. In addition, determining the number of increments allows the project team to identify a critical path, otherwise known as the minimum amount of time needed to complete the project. Moreover, with each increment, the project team can develop and update the following requirements phase deliverables: requirements specifications, functional specifications, and acceptance test specifications. These documents provide the basis for carrying out design validation, design review, and design and development planning.

Design Phase. In the design phase, the project team uses the deliverables and other products of the requirements phase as the input for processes that produce the design phase deliverables. Figure 3 illustrates how the application of Agile practices

enhances production of the design phase deliverables. Specifically, the project team uses Class-Responsibilities-Collaborators (CRC) cards to analyze the prioritized product backlog, requirements specifications, functional specifications, and acceptance test specifications. CRC cards have been used in XP projects as a means for creating the design which the project team uses to build the software (Kaur, Choudhary, & Mehta, 2012; Mirakhorli, Rad, Ailee, Pazoki, & Marakhorli, 2008). They have proved an exceptional tool for creating both conceptual and detailed designs. The common language used on CRC cards also permits the involvement of laypeople unfamiliar with programming terminology. The project team uses CRC Cards instead of other Agile practices such as emergent design because the CRC cards lend themselves better to conforming to FDA design controls.

The CRC cards provide the raw materials for creating preliminary design specifications, detailed design specifications, interface design specifications, and software test plan specifications documents. Creation of these documents occurs during the first iteration of the product lifecycle, yet the project team updates them in each subsequent iteration. Including the documents associated with a traditional Waterfall approach permits conformance with FDA design controls.

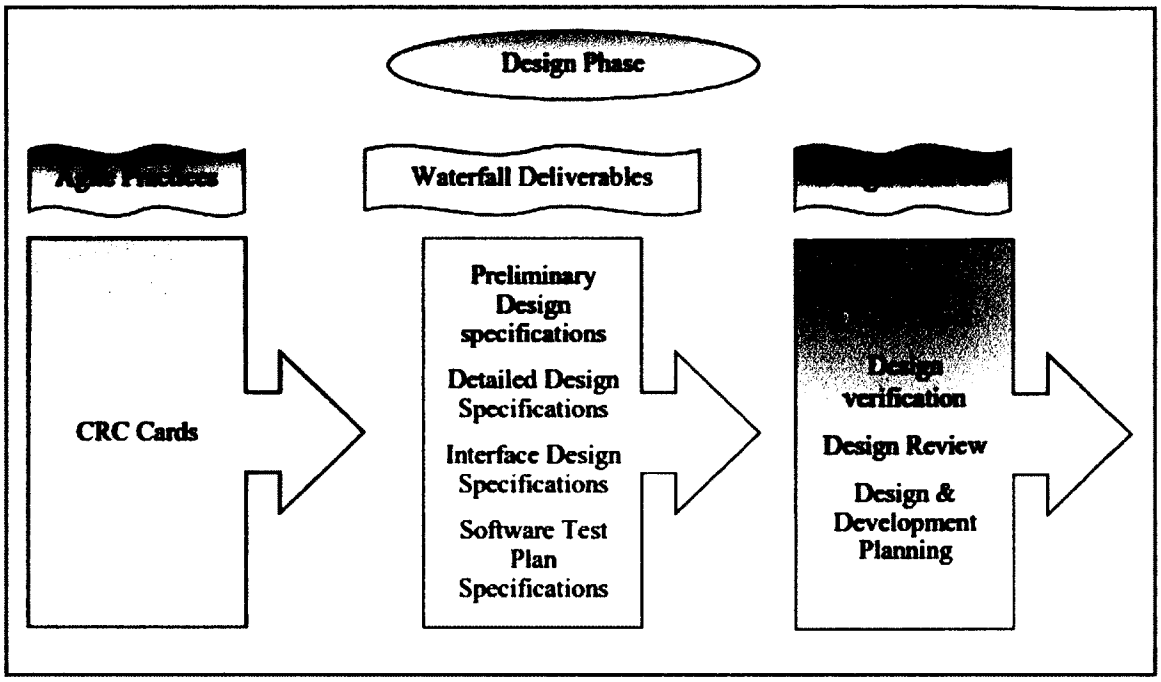


Figure 3. The design phase of the Agile augmented Waterfall model.

Implementation and Testing Phase. One of the key advantages to implementing Agile practices in a Waterfall environment is that testing goes from occurring all at the end of design efforts to being an integral part of the design, development, and coding process (Cao & Ramesh, 2008; McCormick, 2012). As such, the Agile augmented Waterfall model—shown in figure 4—contains an implementation and testing phase where the unification of application coding and testing processes results in a single enhanced phase. The enhanced phase offers several benefits for testing. First, there is constant monitoring of whether or not the product meets design inputs. In addition, testing becomes the responsibility of everyone on the team as opposed to just testers. Finally, bug fixes occur instantaneously as opposed to at the end of coding. These factors

combine to produce applications of an enhanced quality level, meaning greater customer satisfaction.

As the basis for the enhanced implementation and testing phase, there is a daily stand-up meeting that lasts approximately fifteen minutes. These meetings function to enhance internal and external communication, ensuring that all stakeholders are aware of information relevant to the design process (Pikkarainen, Haikara, Salo, Abrahamson, & Still, 2008). In addition, the project team reviews the sprint backlog—a list of tasks identified for completion during the given sprint—at this point. In addition to the daily stand-up, the project team utilizes incremental development whereby implementation and testing occurs via increments and iterations. As the project team works on each increment, it employs pair programming, which provides the added benefits of enhanced code quality, increased level of code understanding amongst the team, constant code reviews, and improved testing and debugging (Begel & Nagappan, 2008). As these pairs work on completing the code, they employ test-driven development (TDD), refactoring, and context-driven testing in order to realize a higher code quality. Finally, a commitment to continuous deployment of a working application must be the end-goal of this phase. This facilitates greater levels of testing, not to mention an opportunity for customers to provide valuable feedback.

During this process, these Agile implementation and testing practices traditionally yield limited documentation. To conform to FDA design controls, the project team must work beyond the expectations of Agile practices to create the following Waterfall deliverables: source code listing, unit test report, sub-system test report, and acceptance

test report. These documents combined with the source code and a completed application version permit completion of activities associated with design output, design verification, design review, and design and development planning.

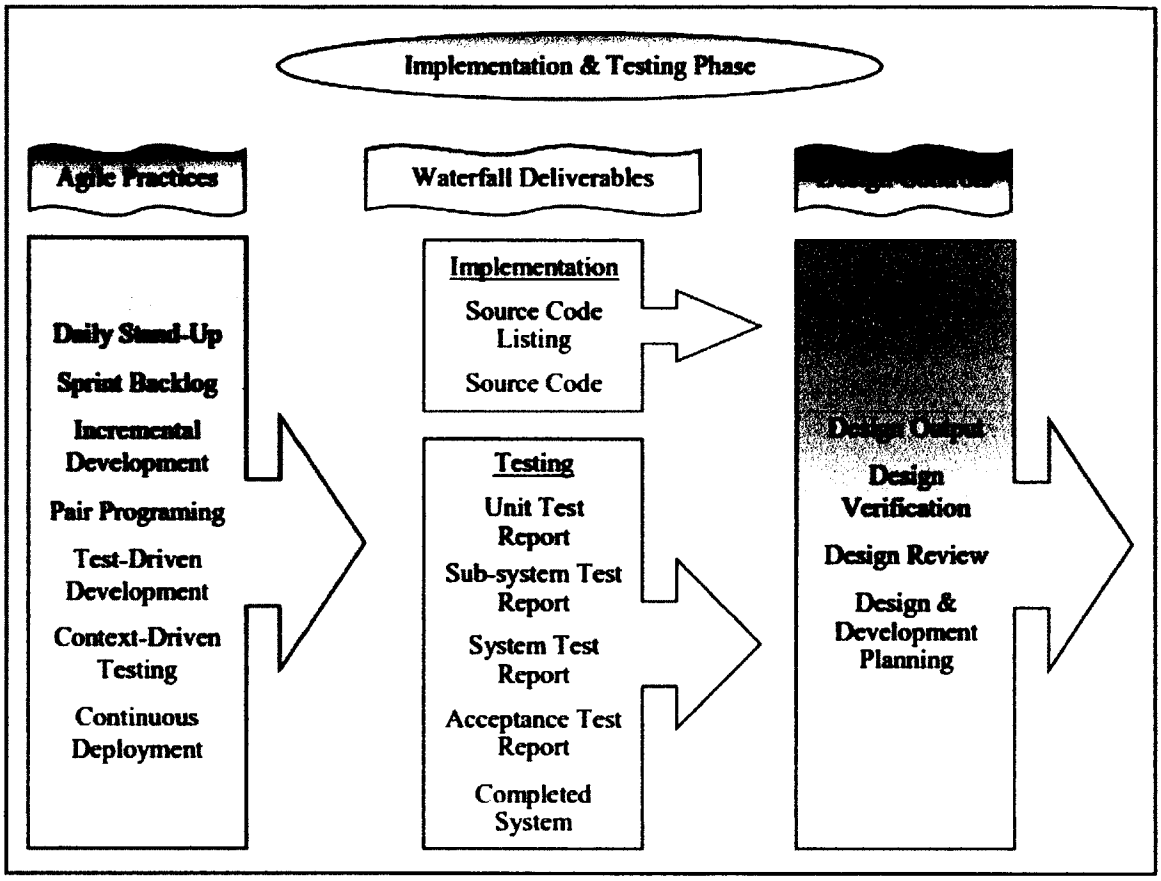


Figure 4. The implementation and testing phase of the Agile augmented Waterfall model.

Deployment Phase. The deployment phase is the final phase of the Agile augmented Waterfall model. As shown in figure 5, the deployment phase carries on the use of continuous deployment and incorporates an iteration or milestone retrospective. As noted above, continuous deployment allows the organization to gain valuable customer feedback. However, the completion of a functional mobile medical application

does not necessarily translate into a release to the public. Rather, the organization only releases the product of specified increments to the public. The project team releases other completed functional mobile medical applications as beta versions that reach only a select few customers. In addition, a functional mobile medical application delivers value to the organization, and the more often that the project team creates this value, the better (Agile Alliance, 2013).

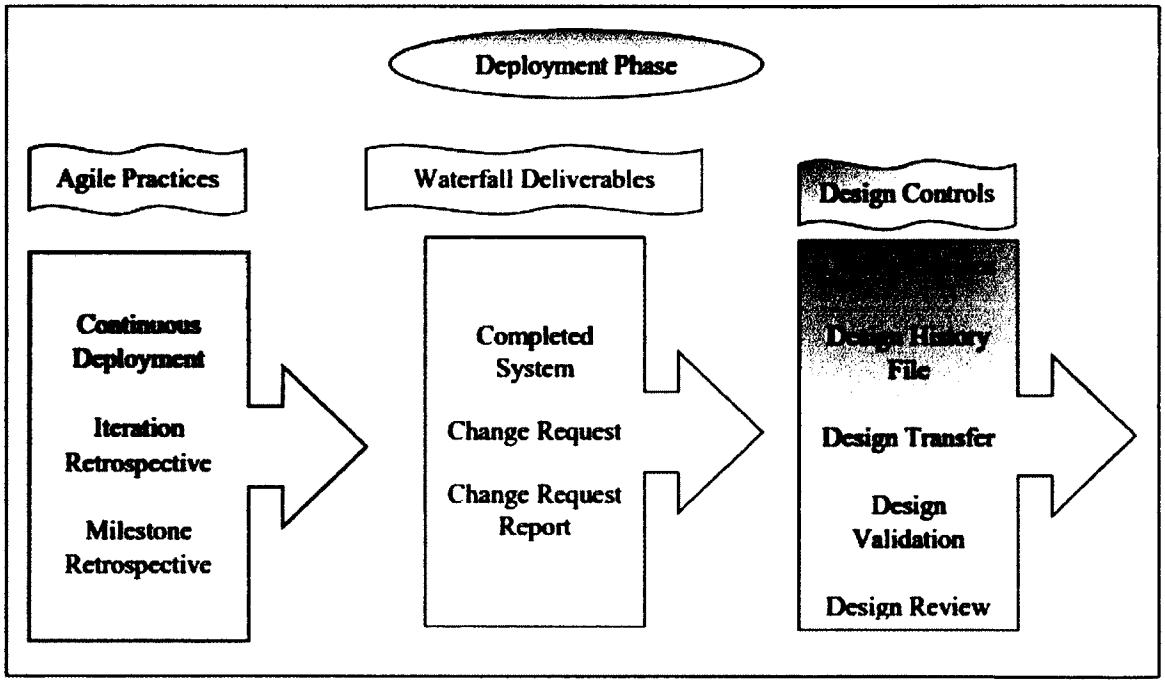


Figure 5. The deployment phase of the Agile augmented Waterfall model.

Additionally, the project team uses both iteration retrospectives and a milestone retrospective during the deployment phase. The retrospectives present an opportunity to address problems with the team and plan improvement strategies (Agile Alliance, 2013). In addition to holding these retrospective meetings, the project team must devote extra

time to complete design validation, design review, and design and development planning. Finally, the team must start a DHF and begin keeping track of design changes.

Fulfilling Market Requirements

In addition to facilitating the realization of Waterfall deliverables and conformance to FDA design controls, it is important that the Agile augmented Waterfall model also orients the organization towards market competitiveness. To do this, the augmented Waterfall model must fulfill the unique requirements for working in the mobile medical application market. As such, this section provides an overview of how the augmented model fulfills the mobile medical application market needs identified during the literature review.

Multiple Platforms and Devices

As noted earlier, project teams must design mobile medical applications in a manner that is responsive to the need to accommodate various different mobile platforms and devices, as well as constantly updated platforms and devices (e.g. Holder, 2013; Joorabchi et al., 2013; Wasserman, 2010). Table 2 provides a list of Agile practices chosen for usefulness in addressing this problem, as well as an explanation of how the Agile practice addresses the specific problem.

Table 2

Agile Practices for Addressing the Need to Design for Multiple and Changing Platforms and Devices

Agile Tools for Addressing the need to Design for Multiple and Changing Platforms and Devices	**Explanation of Augmentation**
Incremental and Iterative Development Process	Through applying an incremental and iterative design process, the project team can adjust increments through a formal process to respond to multiple and changing platforms and devices.
Product Owner	Having a product owner on-site ensures that the project team remains current on changes in requirements due to different or updated platforms and devices. The product owner also helps the team better decide to which platform and device changes they should respond.
Requirements Prioritization	The requirements prioritization functions as a means for deciding and organizing which platform and device requirements the project team will respond to first.
Product Backlog	The product backlog serves as an extension of requirements prioritization by identifying which devices and platforms the project team will design for, as well as when they will undertake the design process.
CRC Cards	CRC cards help the project team to organize design requirements and product features according to differences in mobile platforms and devices.
Context-Driven Testing	Context-driven testing empowers the project team to identify and address problems applicable to all versions of an application, as well as problems applicable to only a specific platform or device.

Device Network Speed and Security.

Additionally, the literature review yielded the conclusion that variability in cellular bandwidth type, capability, and reliability can significantly affect the performance of an application and the security level of patient privacy (e.g. Dehlinger & Dixon, 2011; Holzer & Ondrus, 2011; Wasserman, 2010). Table 3 provides a list of Agile practices chosen for usefulness in addressing this problem, as well as an explanation of how the Agile practice addresses the specific problem.

Table 3

Agile Practices for Addressing the Need to Design for Variations in Network Capability and Security

Agile Tools for Addressing the need to Design for Variations in Network Capability and Security	**Explanation of Augmentation**
Incremental and Iterative Development Process	An incremental and iterative development process ensures that the project team does not design more than is necessary at a given time. This results in greater attention to detail and higher product quality, meaning there is greater attentiveness to variability in network capability and security.
User Stories	User stories provide a window into how, where, when, and why a mobile medical application may be used. As such, the product team can better estimate parameters for network capability and security levels.
Context-Driven Testing	Context-driven testing works to address changes in network environments. Both automated and manual context-driven testing help ensure that a mobile medical application works under the parameters of different networks.
Continuous Deployment	Continuous deployment provides a working mobile medical application at the end of each iteration. As such, the project team can begin to understand what changes must be made to adapt the application to different network environments.

Unparalleled Customer Quality Requirements

Many researchers and industry professionals describe the mobile medical application market as having extremely high demands for quality (e.g. Andjelkovic & Imaizumi, 2012; Holzer & Ondrus, 2011; Nicol, 2013). As such, the augmented model proposed in this thesis employs Agile practices in order to increase quality levels. Table 4 lists the Agile practices relevant to meeting this goal, as well as an explanation of how each practice is useful.

Table 4

Agile Practices for Addressing the Need to Meet Demands for Product Quality

Agile Tools for Addressing the need to Meet High Demands for Product Quality	Explanation of Augmentation
Incremental and Iterative Development Process	An incremental and iterative development process ensures that the project team does not design more than is necessary at a given time. This results in greater attention to detail and higher product quality.
Requirements Prioritization	The project team uses requirements prioritization as a means for deciding and organizing work on features and product requirements. The requirements prioritization facilitates incremental and iterative development, hence increasing quality.
Pair Programming	Pair programming ensures a higher level of code quality because one team member constantly checks and confirms another team member's work. This ensures quality directly by yielding higher quality code, as well as indirectly by providing opportunities for programmers to learn from each other.
Test-Driven Development	Test-driven development increases quality levels by facilitating the development of tests to identify problems with a feature before development of the feature begins. Moreover, TDD ensures that all project team members are testers, which results in an increase in product quality.
Context-Driven Testing	Context-driven testing works to address changes in the environment in which the application is used. Context-driven testing assists the project team in reaching higher levels of product quality through helping to identify unforeseen problems.
Iteration and Milestone Retrospectives	Iteration and milestone retrospectives improve quality indirectly by helping the team to identify strengths and weaknesses, as well as action plans for helping to address the weaknesses.

Short Product Development Cycles

Nicol (2013) notes that the mobile application market requires relatively short product design lifecycles, and Agile practices help to break the product lifecycle up into iterations that yield working software. Table 5 provides a list of Agile practices chosen for usefulness in addressing this problem, as well as an explanation of how each Agile practice addresses the problem.

Table 5

Agile Practices for Facilitating Shorter Product Development Lifecycles

Agile Tools for Facilitating Shorter Product Development Lifecycles	**Explanation of Augmentation**
Incremental and Iterative Development Process	An incremental and iterative development process ensures that the project team does not design more than is necessary at a given time. This permits the development of shorter product development lifecycles and the delivery of working products more regularly.
Requirements Prioritization	The requirements prioritization functions as a means for deciding and organizing requirements in order of completion based upon priority. The requirements prioritization facilitates incremental and iterative development, thus facilitating shorter product development lifecycles.
Continuous Deployment	Integral to practicing Agile principles, the presence of continuous deployment illustrates the true realization of Agile practices. This means that the organization consistently produces working mobile medical applications that may or may not be released to the public.

Testing Difficulty

Testing emerged in the literature review as a particularly challenging characteristic of the mobile medical application market (e.g. Kumar & Chauhan, 2013; Muccini et al., 2012; Wasserman, 2010; Zahra et al., 2013). As such, Table 6 lists several Agile practices helpful for responding to this market characteristic.

Table 6

Agile Practices for Addressing Testing Difficulty

Agile Tools for Addressing Testing Difficulty	Explanation of Augmentation
Test driven development	Test-driven development helps address the testing difficulty associated with mobile medical applications because it makes testing the responsibility of everyone on the project team as opposed to just the testers. Moreover, TDD requires designers and software developers to develop tests for a product or feature prior to designing it, thus ensuring that testing is the first consideration when designing the application.
Context-driven testing	Context-driven testing works to address changes in the environment in which the application is used. Context-driven testing assists the project team with identifying hard to notice problems.
Continuous deployment	Continuous deployment means that several working versions of an application are available before finishing the final application. This positions the organization to start testing early and often.

Case Study: Applying the Agile augmented Waterfall Quality System to the iOxyMonitor

In the proposed QS, it is clear that the application of Agile practices facilitates conformance to FDA design controls, assists in realization of Waterfall phase deliverables, and orients an organization towards competitiveness in the mobile medical market. Having completed the presentation of the proposed QS, the subsequent step in this research methodology is to demonstrate application of the Agile augmented Waterfall model to the actual design and development of a medical device. Thus, this section provides a detailed illustration of the application of the Agile augmented Waterfall model to ABC Health Technologies' design and development of the iOxyMonitor, a mobile medical application that works in conjunction with a peripheral device to measure oxygen saturation in a patient's blood. The case study clearly

delineates how the application of Agile practices aligned to corresponding Waterfall phases and deliverables creates an enhanced design and development process for mobile medical applications.

The author used real data provided by ABC Health Technologies to construct this case study. The case study represents a hypothetical situation in which the organization has fully applied the Agile augmented Waterfall model and corresponding Agile practices. This provides a thorough understanding of the application of the proposed QS in a practical scenario. Though the author used the organization's records and documentation for the iOxyMonitor for the development of this thesis, the author has maintained the company's confidentiality by using pseudonyms for the organization, device names, and any other identifying information.

Through reviewing this case study, it is overwhelmingly clear that in a practical scenario, applying Agile practices to create an enhanced Waterfall QS successfully resulted in a superior design and development process that ensures conformance to FDA design controls. Moreover, the practical benefits in terms of satisfying customers and orienting a company towards market competitiveness also justify application of the proposed QS to future design and development processes for mobile medical applications.

iOxyMonitor Requirements Phase

In the requirement phase, the project team at ABC Health Technologies worked to develop a complete picture of how the iOxyMonitor application would behave. This required a complete listing of both functional and non-functional requirements. In order

to accomplish this task, the project team first identified all primary stakeholders and customers for the iOxyMonitor application. The primary stakeholders at ABC Health Technologies and its subsidiaries were the manufacturing, distribution, sales, and maintenance departments while the primary customers were physicians, nurses, patients, and care providers. The activities that worked to translate stakeholder and customer input into both functional and non-functional requirements demonstrated application of Agile practices in pursuit of the requirements phase deliverables.

All product team members either participated in interviews with the primary stakeholders and customers or utilized data from said interviews to compose user stories. Table 6 provides a sample of the user stories developed through this work. These user stories were important because they described both functional and non-functional requirements in a manner that maintained the principle that a working application equates to business-value. Moreover, the user stories lead to the development of features in small enough chunks to facilitate requirements prioritization, which in turn set the foundation for an iterative and incremental workflow.

Upon completion of the development of user stories, the project team met to develop a project vision. Table 7 shows the project vision for ABC Health Technologies' iOxyMonitor. The project vision served as an overarching goal for project design and development that everyone on the project team shared. Essentially, it kept at the forefront of everyone's thoughts an understanding of where the team was going with the project and what the mobile application would look like when the team got there.

Table 7

Sample of User Stories Developed for the iOxyMonitor

Stakeholder or Customer	User Story
Patient	As a patient, I can learn how to use the application intuitively so that I do not need to invest considerable time into learning how to use the application.
Distributor	As a distributor, I want the application to have a software footprint of less than 50 megabytes so that it can easily and quickly be distributed and downloaded via the numerous application marketplaces without requiring a Wi-Fi connection.
Physician	As a physician, I can use the application with minimal power consumption so that I can use it all day long without recharging my phone.
Sales	As a sales person, the peripheral device that lends functionality to the application must cost less than 20 USD so that the device is financially accessible to a wide range of users.

Table 8

iOxyMonitor Project Vision Statement

iOxyMonitor Project Vision Statement
For physicians, nurses, patients, and care providers who need to take and track blood oxygen saturation levels in a cost efficient manner that can be used in a variety of settings, the iOxyMonitor application is a pulse oximeter that allows customers to use a mobile phone or device in conjunction with a peripheral device to measure, track, and transmit blood oxygen saturation levels. Unlike traditional oximeters, our product does not require a significant hardware investment.

Using the user stories and the product vision, the team then developed a product backlog and used requirements prioritization to sort the product requirements into sprints. Table 8 shows a portion of the product backlog and the results of the requirements prioritization. Note that the product backlog is a collection of all user stories, yet it also includes information regarding how high of a priority the associated feature is, the sprint that it will be included in, and the story points associated with completing the feature. The priority column lists the results of requirements prioritization. The number of story points is an arbitrary measure used to gauge the effort required to complete the given feature relative to other items on the product backlog.

The "added in sprint" column is integral to the incremental and iterative development process mentioned earlier in this chapter. As noted, in the Agile augmented Waterfall model, design and development occurs in an incremental manner, which means that the project team does not develop the entire product at the same time. Rather, there are iterations or sprints through the phases during which the project team works on only selected features and maintains the goal of developing a working mobile application by the end of each sprint. Hence, the initial requirements prioritization helped the project team at ABC Health Technologies divide product features into six different sprints. The project team included the most important features in the first sprint and the least important features in the last sprint. This column tells in which sprint each product feature would be included. Moreover, the team used the number of sprints to determine a critical path of thirty-six weeks. The project team based this figure on the six identified sprints and the need for at least six weeks to complete each sprint. The remainder of this

case study details the work carried out during only the first sprint for this project at ABC Health Technologies.

As a final note regarding Agile practices used in the requirements phase, the project team utilized the input of physicians, patients, nurses, and caregivers. While it would have been difficult to keep all of these prospective customers on-site throughout the entire design and development process, the organization paid a registered nurse to remain on-site for the entire design and development process, exemplifying the importance that product owners play in an Agile augmented QS.

Through utilizing these Agile practices, the project team enhanced the requirements phase for the iOxyMonitor so as to better capture and prioritize customer and business requirements related to the design and development process. The project team simultaneously supported the following FDA design controls: design validation, initial design and development planning, and design review. The project team did this through remaining committed to realizing the following traditional Waterfall requirements phase deliverables: software development plan, requirements specifications, functional specifications, and acceptance test specifications.

Table 9

iOxyMonitor Product Backlog and Requirements Prioritization Results

1	High	Function	physician, nurse, patient, or care provider	be able to measure blood oxygen saturation levels	I can monitor patient health	1	100
2	High	Operation	patient	learn how to use the application intuitively	I do not have to invest a significant amount of time or training into learning how to use the application	1	50
3	Medium-High	Characteristics	distributor	the application to have a software footprint of less than 50 megabytes	it can easily and quickly be distributed and downloaded via the numerous application marketplaces without requiring a Wi-Fi connection.	1	50
4	Medium	Behavior	physician	use the application with minimal power consumption	I can use it all day long without recharging my phone.	2	25
5	High	Characteristics	salesperson	have a peripheral device which lends functionality to the application that costs less than 20 USD	the device is financially accessible to a wide range of users.	1	75
6	Medium	Function	physician	be able to track and plot on a graph a patient's blood oxygen saturation levels	I can track progress throughout the course of treatment	3	50
7	Medium-Low	Function	physician, nurse, patient, or care provider	transmit blood oxygen saturation level data from the mobile device to computers	data can be stored in a patient's electronic file.	5	100

IOxyMonitor Design Phase

ABC Health Technologies applied the Agile tool, CRC cards, to help translate design inputs into design outputs during the design phase for the iOxyMonitor. Figure 6 shows the flow of design inputs to design outputs. As shown, the project team first translated information from the design and development plan, requirements specifications, functional specifications, and the prioritized product backlog into CRC cards. The CRC cards functioned as a tool for creating preliminary design specifications, and later, detailed design specifications and interface design specifications. Additionally, members of the project team also used the software development plan and the acceptance test specification to develop software test plan specifications.

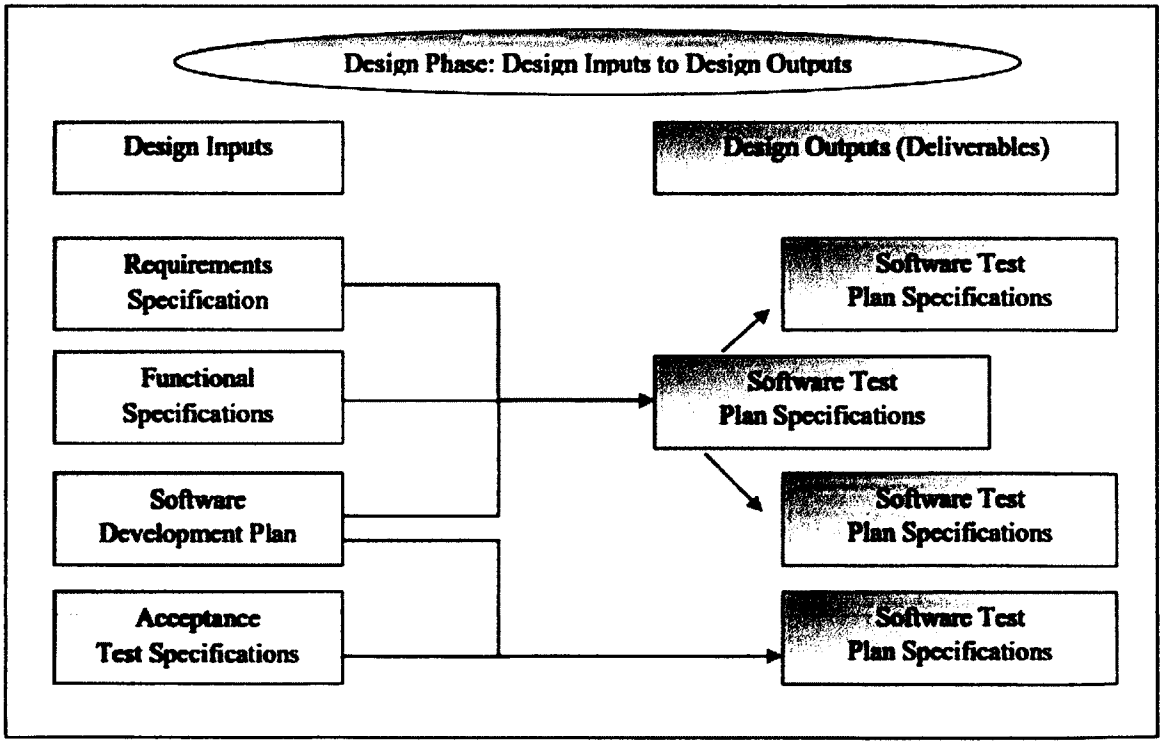
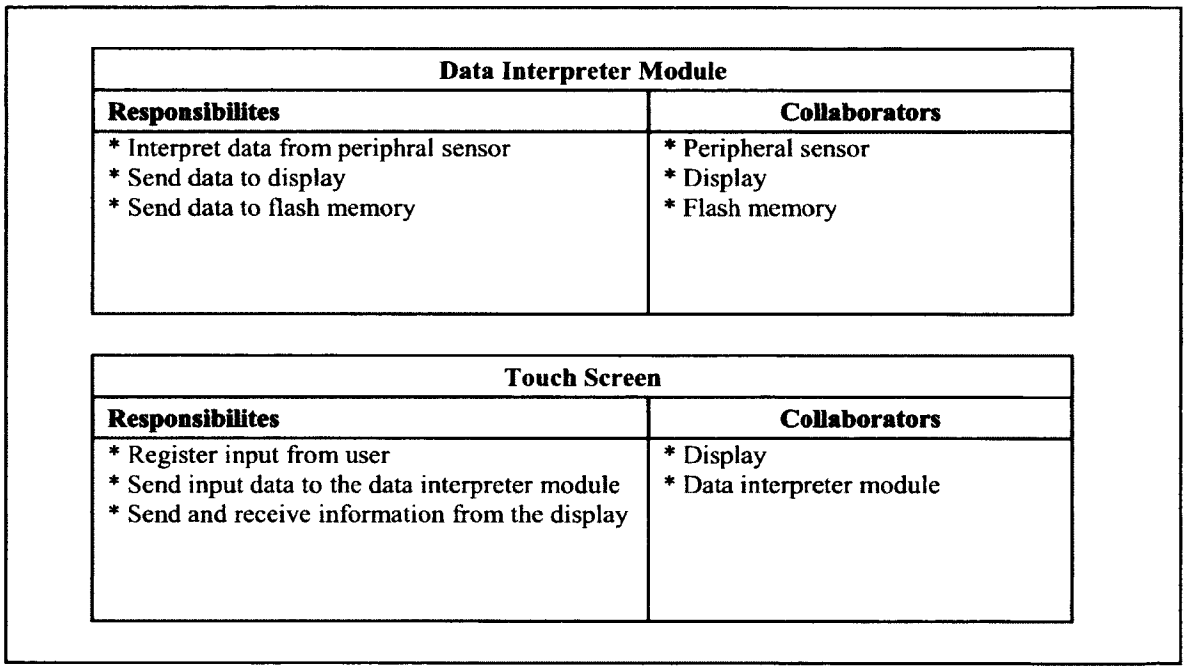


Figure 6. Translation of design inputs into design outputs during the design phase.

On a more detailed level, the project team created CRC cards to model scenarios that may play out when using the mobile medical application. Specifically, each CRC card represents an entity or class of the entire mobile application and the associated features of the mobile device. Examples of such are the flash memory, the display, and an application-specific data interpreter module. Figure 7 shows some examples of CRC cards that the project team at ABC Health Technologies used during the design and development of the iOxyMonitor while figure 8 shows how the project team used CRC cards to map one of the sub-systems for the iOxyMonitor.



Data Interpreter Module	
Responsibilites	**Collaborators**
* Interpret data from periphral sensor * Send data to display * Send data to flash memory	* Peripheral sensor * Display * Flash memory

Touch Screen	
Responsibilites	**Collaborators**
* Register input from user * Send input data to the data interpreter module * Send and receive information from the display	* Display * Data interpreter module

Figure 7. Pictures of CRC cards used during the design phase for the iOxyMonitor.

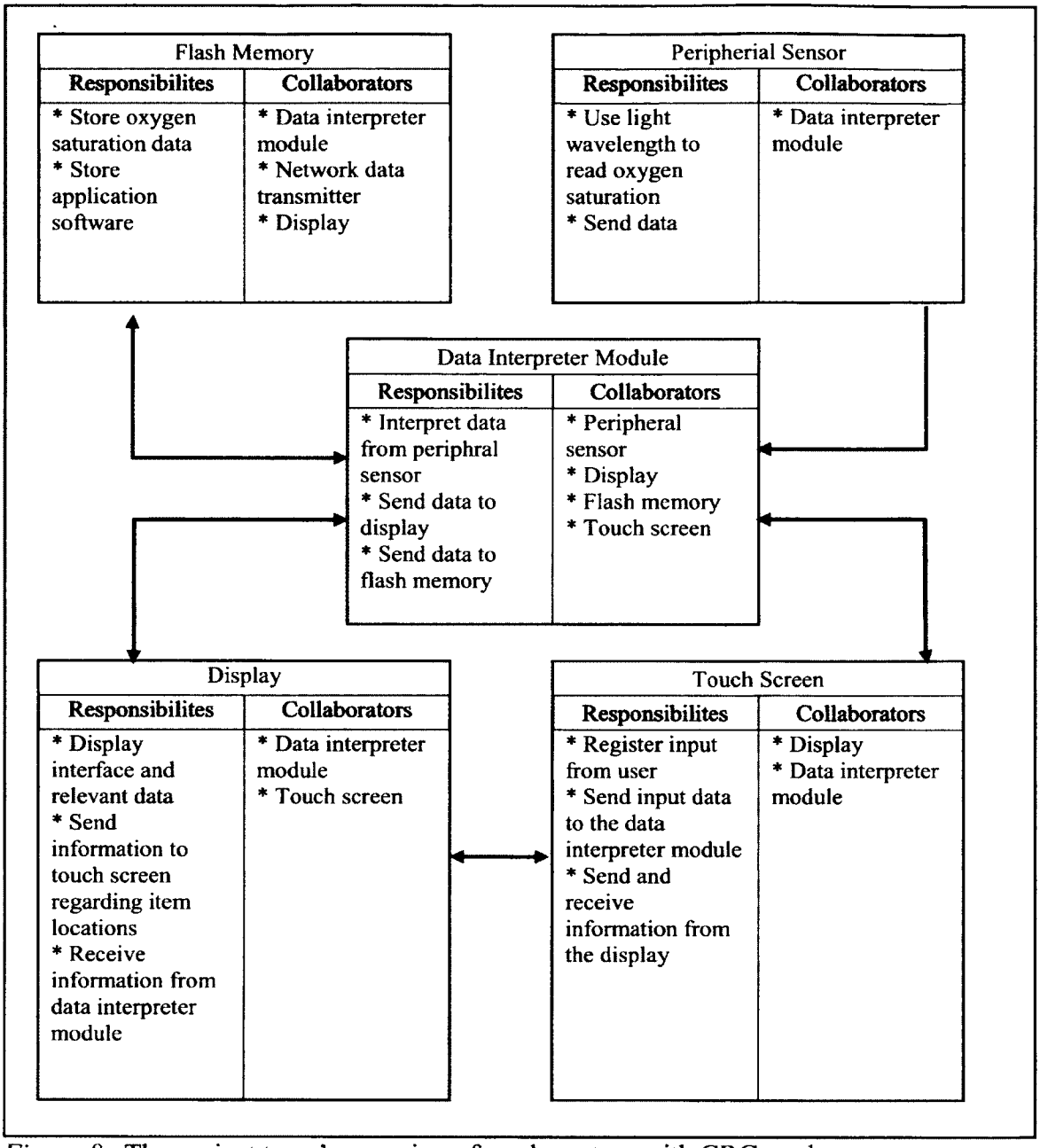


Figure 8. The project team's mapping of a sub-system with CRC cards.

Upon completing the CRC cards, the project team engaged in CRC sessions. These sessions provided a verbal opportunity to review scenarios in which a customer would complete an operation with the mobile application. Through reviewing the scenarios, the project team identified process weaknesses and developed superior design alternatives.

For the project team at ABC Health Technologies, utilizing CRC cards for the design of the iOxyMonitor resulted in an enhanced design phase that more accurately identified preliminary design weaknesses and facilitated development of superior design alternatives. Concurrently, ABC Health Technologies' project team supported the following FDA design controls: design input, design verification, design and development planning, and design review. This occurred because the project team remained devoted to completing the following documents: preliminary design specifications, detailed design specifications, interface design specifications, and software test plan specifications.

iOxyMonitor Implementation and Testing Phase

Upon completing the design phase for the iOxyMonitor, the project team at ABC Health Technologies set out to complete a unified implementation and testing phase. Specifically, the team implemented the following Agile practices and tools: a daily stand-up, incremental development, pair programming, a sprint backlog, TDD, and context-driven testing. Implementation of these tools while still maintaining a commitment to developing the traditional Waterfall documentation ensured conformance to the following

FDA design controls: performing design verification, identifying design outputs, conducting design reviews, and updating the design and development plan.

Specifically, the project team used the Agile tools and practices as a means for translating detailed design specifications, interface design specifications, and software test plan specifications into a source code, a source code listing, unit test reports, sub-system test reports, a system test report, and an acceptance test report. Figure 9 displays the application of Agile tools and practices to the implementation and testing phase of the iOxyMonitor.

As shown in Figure 9, all development during the implementation and testing phase required at least two software developers working together. This pair programming meant that one developer at a workstation did the actual programming while the other developer watched, learned, and made suggestions. Such a style resulted in a higher quality code, meaning that there was less rework in the end.

As the programmers worked, they also increased the quality of the code by employing TDD, refactoring, and context-driven testing. Using TDD meant that the software developers began their work by developing a test for a piece of functional code before even writing the functional code. Once the team developed the test, the pair wrote just enough functional code to ensure that the test ran but failed. Then they ran the tests. The purpose of this was to test the test. If the very low-level functional code failed the test, then the developers knew that the test had worked for identifying faulty code.

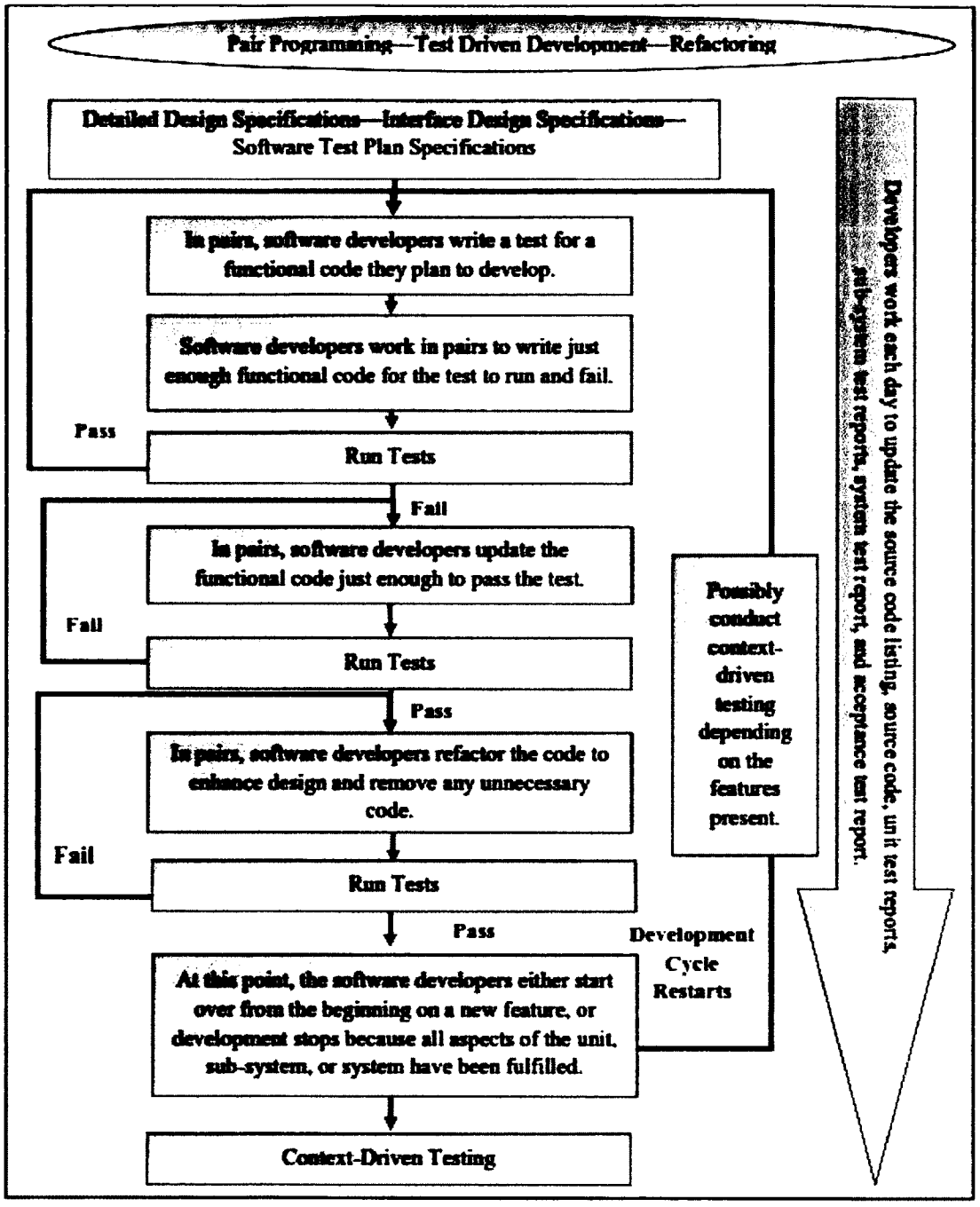


Figure 9. Application of Agile tools and practices to the implementation and testing phase of the iOxyMonitor.

Once the software developers completed the above process, the software developers wrote just enough functional code to pass the test. Following passage of the test, the software developers then worked on refactoring the code. This meant that they worked to enhance the design while removing any duplication. The software developers then ran tests once again. Passage ensured that they had not damaged code functionality while refactoring.

At this point, software developers resolved to do one of two things. The first and more likely possibility was that they would return to the beginning of the cycle to add yet another part of the functional code. However, before returning to the beginning, they may or may not have engaged in context-driven testing. Context-driven testing occurred when they had a working application. During context-driven testing, the project team used the application to simulate conditions when customers might use it. For example, team members may have used the application in conditions where there is limited network connection. This type of testing helped to identify problems and bugs that may not have appeared during testing in optimal conditions. While this was one option following refactoring, the other option was to engage in context-driven testing and then stop development because a unit, sub-system, or system was complete.

Throughout this process, software developers worked off a sprint backlog, which consisted of all user stories identified as part of the first sprint during requirements prioritization. Moreover, during implementation and testing, software developers devoted time each day to updating documents. Thus, the project team realized the following deliverables for the Waterfall implementation and testing phases: a source code

listing, a source code, unit test reports, sub-system test reports, a system test report, and an acceptance test report.

Realization of these deliverables permitted the project team to identify formal design outputs and conduct design verification and design review activities. This means that the project team ensured correct alignment of the formal design inputs with the design outputs. In addition to supporting conformance to FDA design controls, the project team developed an enhanced working mobile application as a result of applying Agile tools and practices.

iOxyMonitor Deployment Phase

With the implementation and testing phase of the first sprint complete, ABC Health Technologies had completed the first working version of the iOxyMonitor. This signaled an opportunity to display the project team's commitment to the Agile principle of continuous deployment, as well to conduct an iteration retrospective. Moreover, this phase provided an opportunity to secure the following Waterfall deliverables: documentation of design changes in a change request report and delivery of a completed system. Use of these Agile practices and realization of these Waterfall deliverables permitted conformance to several FDA design controls.

Having delivered a completed system in the form of a working mobile medical application, ABC Health Technologies then set out to deploy the first version of the iOxyMonitor into a customer environment rather than deploy it to the general market. Though only a deployment of an application with limited functionality, this beta release

provided an opportunity to identify problems. The project team could then work on these problems during later iterations.

Aside from deployment, having a completed system at the conclusion of the iteration permitted the project team to conduct activities associated with the following FDA design controls: design validation, design review, design changes, design and development planning, design transfer, and DHF. When conducting design validation, the project team ensured that the working application conformed to the user needs identified for fulfillment during the first sprint. The project team had specified these user needs during the requirements phase. Continuing, the team conducted design review activities to ensure that design processes occurred in accordance with the design and development plan. At this time, the project team also updated the design and development plan to reflect any changes to the original plan. They documented and approved such changes through the design change process. The project team then conducted initial activities associated with design transfer; however, a more thorough design transfer process would take place later when the team planned to release the completed system to the general market. As a final point, the team gathered all documentation from the first iteration and compiled it in a formal DHF. Through these processes, the project team thoroughly fulfilled the FDA design control requirements.

To maximize returns in the form of an enhanced design and development process for the iOxyMonitor, the project team completed an iteration retrospective during the deployment phase as well. This allowed the team to evaluate itself by first assessing whether the team completed the goals of the first sprint; specifically, the team determined

whether they had fulfilled all user needs identified for completion during the first sprint. Additionally, the team evaluated the process by addressing what went well and what did not. The findings of this assisted the team as it moved into the second iteration.

Having reviewed the deployment phase for the first iteration of the design and development process for the iOxyMonitor, it is clear that ABC Health Technologies enhanced the design process by adhering to the principle of continuous deployment and implementing an iteration retrospective. Furthermore, employing continuous deployment enabled the team to realize the following Waterfall deliverables: delivery of a completed system and completion of a change request report. This in turn supported the project team's activities for ensuring conformance to the FDA design controls.

Rationalizing the Use of the Agile Augmented Waterfall Quality System through its Application to the iOxyMonitor

While ABC Health Technologies' original design process adhered to a strict Waterfall model to ensure conformance to FDA design controls, this case study illustrates an adaptation of the organization's design and development activities that yields an enhanced design and development process. This improved design and development process functions as a result of aligning Agile tools and practices to Waterfall deliverables and FDA design controls. Thus, the proposed Agile augmented Waterfall model provides a structure for applying Agile practices, tools, and principles in the FDA regulated environment in which designers of class III mobile medical applications must work.

Moreover, the case study emerges as a noteworthy contribution to QA for mobile medical applications because it supplies an enhanced design and development process that organizations can use for a variety of mobile medical applications. This enhanced process simultaneously ensures conformance to FDA design controls while also orienting an organization towards the goal of market competitiveness. This occurs because the Agile augmented Waterfall model delivers the following benefits:

- an enhanced ability to design for multiple and changing platforms and devices;
- an enhanced ability to design for variations in network capability and security;
- an enhanced level of quality that more completely fulfills customer expectations;
- an ability to deliver a working application prior to completing the entire design process; and
- a greater ability to test the mobile application more completely and appropriately.

CHAPTER 4

RESULTS AND DISCUSSION

This section analyzes the benefits of applying Agile tools and practices to the Waterfall model for the purpose of designing and developing mobile medical applications. The literature review established the need for applying Agile to mobile medical applications through showing that the Waterfall method falls short of fulfilling the market and design and development needs of mobile medical applications. In addition, the literature review showed that Agile lacks the lifecycle, clear milestones, defined processes, and documentation necessary to ensure conformance to regulatory requirements on its own when designing mobile medical applications. This set the course for developing a QS based upon the alignment of Agile tools, practices, and principles with Waterfall phase deliverables. Moreover, the case study illustrated the practicality of applying the proposed Agile augmented Waterfall model to the design and development of mobile medical devices. Taken together, the elements of this thesis establish the basis for promoting implementation of the proposed Agile augmented Waterfall model for the design and development of mobile medical applications.

Specifically, application of Agile tools, practices, and principles through their alignment with the Waterfall phase deliverables enhances the traditional Waterfall design and development process for mobile medical applications in the following manners:

- increasing responsiveness to changing product requirements through implementing shorter product development lifecycles, incremental development, and multiple opportunities to alter design requirements;

- enhancing the ability to design for variability in device capability, as well as network capability and security levels through employing context-driven testing;

- augmenting the design and development process to ensure greater customer satisfaction and operational efficiency through implementing an incremental development process, requirements prioritization, pair programming, TDD, context-driven testing, and iteration and milestone retrospectives;

- increasing business value via instituting shorter product development cycles that yield working software that can either be released on a beta level for testing or on a distribution level to the general market; and

- increasing the effectiveness of product testing through implementing continuous deployment, TDD, and context-driven testing.

Continuing, research establishes that these enhancements to the design and development process deliver value to the company. First, Agile-related enhancements result in a reduction in requirements waste and change requests, which in turn results in greater operational efficiency (Petersen & Wohlin, 2010). Likewise, these enhancements result in greater code and application quality, which means the organization spends less time and money on fixing defects during both the design and development process and maintenance of the application. This means that the organization spends less money, and customers are more satisfied (Petersen & Wohlin, 2010). Additionally, these enhancements mean that the project team delivers a working application, which translates into business value, on a more regular basis with a reduced time investment (Dyba & Dingsoyr, 2008). This provides the business with an opportunity to release the application to the market or use it as an opportunity to gain valuable feedback. Moreover, these enhancements, with an emphasis on pair programming and iteration and milestone retrospectives, result in learning on the part of team members, better

positioning them to make valuable contributions to the organization at a later point. As a final point, these enhancements mean that there is a diminished level of risk associated with mobile medical application development projects (Dyba & Dingsoyr, 2008). This is due to the increased value and adaptability that the project team delivers early in the project development lifecycle.

As such, it is clear that application of the proposed Agile augmented Waterfall model positively affects the following factors: operational efficiency, business value, product quality, and team member learning. At the same time, there is a reduced level of risk associated with application of the proposed Agile augmented Waterfall model. Consequently, it is a clear that the proposed Agile augmented Waterfall model gives organizations the advantage of allowing them to operate efficiently and competitively in the mobile medical application market while maintaining conformance with FDA regulatory requirements.

CHAPTER 5

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

Summary

The task analyzed in this thesis is the necessity to design and develop mobile medical applications while ensuring conformance to regulatory requirements and responding to the unique needs of the mobile medical application market. Doing so first implies adherence to those requirements set forth in the 2013 FDA published, *Mobile medical applications: Guidance for industry and Food and Drug Administration staff,* and the FDA QS Regulation. Concurrently, organizations must also respond to the following market needs when working on mobile medical applications:

- to design for operation on multiple platforms and devices;
- to design such that the application can operate on networks with varying capabilities and security levels;
- to fulfill incredibly high customer demands for quality;
- to implement shorter product development lifecycles that deliver working applications with greater frequency; and
- to respond to significant testing barriers.

The QA methodologies and tools investigated as a means for addressing the challenges associated with the mobile medical application market were the Waterfall model for software development and Agile tools, practices, and principles for software development. The former was instrumental for ensuring conformance to FDA regulatory

requirements while the latter was instrumental to addressing the unique market needs for the design and development of mobile medical applications.

Through analysis of these quality tools and methodologies, this study provides a framework for applying Agile tools, practices, and principles in unison with a Waterfall model to meet the regulatory and market needs associated with the design and development of mobile medical applications. Specifically, the pursuit of Waterfall phase deliverables through the application of Agile tools, practices, and principles simultaneously ensures conformance to FDA design controls while also increasing operational efficiency, business value, product quality, and team member learning. Application of this framework holds the promise of ensuring that organizations can maintain the profit margins and competition levels necessary for operating in the mobile medical application market.

Contributions

This thesis provides several contributions to the body of research related to QA for mobile medical applications. Specifically, Lin and Fan (2009) and Rasmussen et al. (2009) investigate the applicability of Agile tools, practices, and methodologies in unison with a Waterfall lifecycle to the design and development of medical software. The present thesis extends this research to the arena of mobile medical applications. Moreover, this thesis presents a more comprehensive framework than that posited in either piece of research for aligning Agile tools, practices, and principles to the Waterfall model.

Likewise, research has also investigated the application of Agile tools, practices, and principles to the development of non-medical mobile applications (e.g. Andjelkovic & Imaizumi, 2012; Corral, Sillitti, & Succi, 2013; Nicol, 2013). This literature has underscored the important benefits that application of Agile has yielded for project teams working on mobile applications. Specifically, application of Agile has proven to help the project team respond to the market requirements for mobile applications. Understanding that application of Agile alone may not ensure adherence to regulatory requirements, this thesis extends the research on applying Agile to non-medical mobile applications such that QA professionals can begin to understand how Agile can be applied to mobile medical applications. Hence, it is clear that this piece has made significant contributions to the body of research related to QA for mobile medical applications.

Recommendations

Despite the contributions that this thesis makes to the body of research on QA for mobile medical applications, there still remain several areas for future research. First, there is a rather inconclusive body of research related to the costs associated with implementing Agile tools, practices, and principles and whether or not the costs justify the benefits associated with implementing Agile. There ought to be further research into this area to determine the exact costs associated with implementing an Agile-Waterfall hybrid model.

In addition, where any FDA regulated mobile medical application is concerned, there needs to be a significant amount of time and money invested into ensuring conformance with regulatory requirements. Specifically, there is a vast amount of

documentation created for no other reason than to ensure conformance to FDA design controls. Taking this precedent into consideration, there is research suggesting that perhaps the FDA design controls for medical devices need to be updated to more accurately reflect the current tools available and tasks at hand (e.g. Mitka, 2013; Yetisen et al., 2014). As such, there ought to be more research to determine whether all documentation required for the design and development of mobile medical applications is truly necessary to ensure patient safety. Researchers just might reach the conclusion that the current regulations are outdated.

Finally, the proposed Agile augmented Waterfall model employs a very specific set of Agile tools, practices, and principles in combination with Waterfall phases and deliverables to guide the design and development of mobile medical applications. Moreover, it in no way covers the entirety of Agile tools, practices, and principles available. As such, there ought to be significant research into the issue of what other Agile tools, practices, and principles would be suitable for being included in mobile medical application design and development. As people conduct research on this issue, they ought to direct special consideration towards those tools, practices, and principles that seem particularly adaptable to being included in a Waterfall model.

REFERENCES

REFERENCES

Agile Alliance. (2001). *The Agile manifesto.* Retrieved January 14, 2014, from http://www.agilealliance.org/the-alliance/the-agile-manifesto/

Agile Alliance. (2013). *The Agile Alliance.* Retrieved January 14, 2014, from http://www.agilealliance.org/

Alsmadi, M., Almani, A., & Jerisat, R. (2012). A comparative analysis of Lean practices and performance in the UK manufacturing and service sector firms. *Total Quality Management & Business Excellence, 23*(3-4), 381-396.

Andjelkovic, M., & Imaizumi, S. (2012). Mobile entrepreneurship and employment. *Innovations, 7*(4), 87-100.

Banga, C., & Weinhold, J. (2014). *Essential mobile interaction design: Perfecting interface design in mobile apps*. Upper Saddle River, NJ: Pearson Education.

Barlow, J. B., Giboney, J. S., Keith, M. J., Wilson, D. W., Schuetzler, R. M., Lowry, P. B., et al. (2011). Overview and guidance on Agile development in large organizations. *Communications of the Association for Information Systems, 29*(1), 25-44.

Barton, A. J. (2012). The regulation of mobile health applications. *BMC Medicine, 10*(1), 46-49.

Bassil, Y. (2012). A simulation model for the Waterfall software development life cycle. *International Journal of Engineering and Technology, 2*(5), 1-7.

Begel, A., & Nagappan, N. (2008, October). Pair programming: What's in it for me?. In *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement,* (pp. 120-128). Germany: ACM.

Brewin, B. (2013). *FDA will regulate some mobile medical apps as devices.* Retrieved March 15, 2014, from http://www.nextgov.com/mobile/2013/09/fda-will-regulate-some-mobile-medical-apps-devices/70760/

Cao, L., & Ramesh, B. (2008). Agile requirements engineering practices: An empirical study. *IEEE Software, 25*(1), 60-67.

Cawley, O., Wang, X., & Richardson, I. (2010). Lean/Agile software development methodologies in regulated environments: State of the art. In B. Fitzgerald, K. Conboy, K. Power, R. Valerdi, L. Morgan & K. Stol (Eds.), *Lean enterprise software and systems,* (pp. 31-36). Berlin, Germany: Springer.

Ciaramitaro, B. L. (2011). *Virtual education.* Hershey, PA: IGI Global Snippet.

Cohen, D., Lindvall, M., & Costa, P. (2004). An introduction to Agile methods. *Advances in Computers, 62*, 1-66.

Comstock, J. (2014, August 28). *Bayer's Grants4Apps: An accelerator run by a pharma company.* Retrieved January 14, 2014, from http://mobihealthnews.com/36145/bayers-grants4apps-an-accelerator-run-by-a-pharma-company/

Corral, L., Sillitti, A., & Succi, G. (2013, May). Software development processes for mobile systems: Is Agile really taking over the business? In *2013 1st*

International Workshop on the Engineering of Mobile-Enabled Systems, (pp. 19-24). Germany: IEEE.

Danzis, S. D., & Pruitt, C. (2013). Rethinking the FDA's regulation of mobile medical apps. *SciTech Lawyer, 9*(3), 27.

David, M., & Murman, C. (2014). *Designing apps for success: Developing consistent app design practices*. Burlington, MA: CRC Press.

Davis, B. (2012). *Agile practices for Waterfall projects*. Plantation, FL: J. Ross Publishing.

Dehlinger, J., & Dixon, J. (2011). Mobile application software engineering: Challenges and research directions. *Proceedings of the Workshop on Mobile Software Engineering,* (pp. 29-32). Germany: IEEE.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of Agile software development: A systematic review. *Information and software technology, 50*(9), 833-859.

Dyba, T., & Dingsoyr, T. (2009). What do we know about Agile software development? *IEEE Software, 26*(5), 6-9.

Erickson, J., Lyytinen, K., & Siau, K. (2005). Agile modeling, Agile software development, and extreme programming: The state of research. *Journal of Database Management, 16*(4), 88-100.

Fish & Richardson. (2013). *Wireless medical technologies: Navigating government regulation in the new medical age* [White paper]. Retrieved March 3, 2014, from http://www.fr.com/files/uploads/attachments/FinalRegulatoryWhitePaperWirelessMedicalTechnologies.pdf

Food and Drug Administration [FDA]. (2013). *Mobile medical applications: Guidance for industry and Food and Drug Administration staff.* Retrieved January 15, 2014, from http://www.fda.gov/downloads/MedicalDevices/.../UCM263366.pdf

Gember, A., Akella, A., Pang, J., Varshavsky, A., & Caceres, R. (2012). Obtaining in-context measurements of cellular network performance. *Proceedings of the 2012 ACM Conference on Internet Measurement Conference,* (287-300). New York, NY: ACM.

Gordon, I. (2013). *Managing the new customer relationship: Strategies to engage the social customer and build lasting value.* Mississauga, Ontario: John Wiley & Sons.

Hewlett-Packard Development Company. (2012). *Survival guide for testing modern applications* [white paper]. Retrieved from http://h20195.www2.hp.com/v2/GetPDF.aspx%2F4AA3-5237EEW.pdf

Hoda, R., Noble, J., & Marshall, S. (2011). Supporting self-organizing Agile teams. In G. Cantone, & M. Marchesi (Eds.), *Agile processes in software engineering and extreme programming,* (pp. 73-87). Switzerland: Springer.

Holder, E. (2013). Cloud twin: Interactive cross-platform replay for mobile applications. *Proceedings of the 2013 Companion Publication for Conference on Systems, Programming, & Applications: Software for Humanity,* (pp. 119-120). New York, NY: ACM.

Holzer, A., & Ondrus, J. (2011). Mobile application market: A developer's perspective. *Telematics and Informatics, 28*(1), 22-31.

Istepanian, R. S., Jovanov, E., & Zhang, Y. (2004). Guest editorial introduction to the special section on m-health: Beyond seamless mobility and global wireless health-care connectivity. *IEEE Transactions on Information Technology in Biomedicine, 8*(4), 405-414.

Iversen, J., & Eierman, M. (2013). *Learning mobile app development: A hands-on guide to building apps with IOS and android.* Upper Saddle River, NJ: Pearson Education.

Jeong, Y. J., Lee, J. H., & Shin, G. S. (2008, February). Development process of mobile application SW based on Agile methodology. In *10th International Conference on Advanced Communication Technology,* (Vol. 1, pp. 362-366). Germany: IEEE.

Joorabchi, M. E., Mesbah, A., & Kruchten, P. (2013). Real challenges in mobile app development. In *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement,* (pp. 15-24). Germany: IEEE.

Kaur, M. R., Choudhary, M., & Mehta, M. R. (2012). Agile Process: An Enhancement to the process of software development. *International Journal of Computer Science and Network Security*, 12(7), 101.

Kay, M. (2011). mHealth: New horizons for health through mobile technologies. *Global Observatory for eHealth Series, 3*(1), 1.

Kinsel, D. (2012). Design control requirements for medical device development. *World Journal for Pediatric & Congenital Heart Surgery, 3*(1), 77-81.

Krouse, A. (2012). iPads, iPhones, androids, and smartphones: FDA regulation of mobile phone applications as medical devices. *Ind.Health L.Rev., 9*, 731.

Kumar, M., & Chauhan, M. (2013). *Best Practices in mobile application testing* [White Paper]. Retrieved from http://www.infosys.com/flypp/resources/Documents/mobile-application-testing.pdf

Lee, I. (2012). *Mobile services industries, technologies, and application in the global economy*. Hershey, PA: IGI Global.

Lin, W., & Fan, X. (2009). Software development practice for FDA-compliant medical devices. *2009 International Joint Conference on Computational Sciences and Optimization,* (pp. 388-390). Germany: IEEE.

Lincoln, J. E. (2010). Medical device development under design control. *Journal of Validation Technology, 16*(1), 29.

Martín, D., López-de-Ipiña, D., Alzua-Sorzabal, A., Lamsfus, C., & Torres-Manzanera, E. (2013). A methodology and a web platform for the collaborative development of context-aware systems. *Sensors, 13*(5), 6032-6053.

McCormick, K. A., & Gugerty, B. (2012). *Healthcare information technology exam guide for CompTIA healthcare IT technician and HIT pro certifications*. New York, NY: McGraw-Hill.

McCormick, M. (2012). *Waterfall vs. Agile methodology.* Retrieved February 12, 2014, from http://www.mccormickpcs.com/images/Waterfall_vs_Agile_Methodology.pdf

McGrath, M. J., & Scanaill, C. N. (2013). *Sensor technologies: Healthcare, Wellness and Environmental applications.* Switzerland: Springer.

Mc Hugh, M., Mc Caffery, F., Casey, V., & Pikkarainen, M. (2012), Proceedings from EuroSPI 2012. In *European Systems and Software Process Improvement and Innovation Conference.* Vienna, Austria: Springer.

Miller, K. W., & Voas, J. (2012). Software testing: What goes around comes around. *IT Professional, 14*(3), 4-5.

Millett, S., Blankenship, J., & Bussa, M. (2011). *Pro Agile. NET development with Scrum*. New York, NY: Apress.

Muccini, H., Di Francesco, A., & Esposito, P. (2012). Software testing of mobile applications: Challenges and future research directions. *2012 7th International Workshop on Automation of Software Test (AST),* (pp. 29-35). Germany: IEEE.

Munassar, N. M. A., & Govardhan, A. (2010). A comparison between five models of software engineering. *International Journal of Computer Science Issues, 7*(5), 418.

Murphy, T. E., Duggan, J., Norton, D., Prentice, B., Plummer, D. C., & Landry, S. (2009). *Predicts 2010: Agile and cloud impact application development directions*. Stamford, CT: Gartner.

Newell, D. (2012, June 4). *5 Ways Mobile Apps Will Transform Healthcare.* Retrieved January 14, 2014, from http://www.forbes.com/sites/ciocentral/2012/06/04/5-ways-mobile-apps-will-transform-healthcare/

Nicol, D. (2013). *Mobile strategy: How your company can win by embracing mobile technologies*. Boston, MA: IBM Press.

Omdahl, T. P. (1997). *Quality dictionary.* West Terre Haute, IN: Quality Council of Indiana.

Paetsch, F., Eberlein, A., & Maurer, F. (2012). Requirements engineering and Agile software development. In *2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises,* (pp. 308-308). Germany: IEEE Computer Society.

Perez, S. (2013). *Apple's app store hits 50 billion downloads, 900K apps, $10 billion paid to developers; iTunes now with 575M accounts.* Retrieved June 10, 2014, from http://markets.financialcontent.com/pennwell.photovoltaics/news/read/24389726/apple's_app_store_hits_50_billion_downloads

Petersen, K., & Wohlin, C. (2010). The effect of moving from a plan-driven to an incremental software development approach with Agile practices. *Empirical Software Engineering, 15*(6), 654-693.

Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of Agile practices on communication in software development. *Empirical Software Engineering, 13*(3), 303-337.

Rasmussen, R., Hughes, T., Jenks, J., & Skach, J. (2009). Adopting Agile in an fda regulated environment. *Agile Conference, 2009,* (pp. 151-155). New York, NY: ACM.

Ravindranath, L., Padhye, J., Mahajan, R., & Balakrishnan, H. (2013). Timecard: Controlling user-perceived delays in server-based mobile applications. In

Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles, (pp. 85-100). New York, NY: ACM.

Rowe, J. (2012, December 4). *Mobile Apps Changing Healthcare.* Retrieved February 16, 2014, from http://www.healthcareitnews.com/news/mobile-apps-changing-healthcare

Royce, W., Bittner, K., & Perrow, M. (2009). *The economics of iterative software development: Steering toward better business results*. Boston, MA: Pearson Education.

Scharff, C. (2011). Guiding global software development projects using scrum and Agile with quality assurance. *2011 24th IEEE-CS Conference on Software Engineering Education and Training,* (pp. 274-283). Germany: IEEE.

Scharff, C., & Verma, R. (2010). Scrum to support mobile application development projects in a just-in-time learning context. *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering,* (pp. 25-31). Germany: IEEE.

Smith, G. S. (2013). *Straight to the top: CIO leadership in a mobile, social, and cloud-based world*. Hoboken, New Jersey: John Wiley & Sons.

Srinivasan, J., & Lundqvist, K. (2009). Using Agile methods in software product development: A case study. In *Sixth International Conference on Information Technology: New Generations, 2009,* (pp. 1415-1420). Germany: IEEE.

Sutherland, J., Miller, J., Hollowell, G., Patel, D., & Casanave, C. (1997). *Business object design and implementation: OOPSLA'95 workshop proceedings*. New York, NY: Springer-Verlag.

Svanæs, D., Alsos, O. A., & Dahl, Y. (2010). Usability testing of mobile ICT for clinical settings: Methodological and practical challenges. *International Journal of Medical Informatics, 79*(4), e24-e34.

Takeuchi, H., & Nonaka, I. (1986). The new new product development game. *Harvard Business Review, 64*(1), 137-146.

Taneja, R. K., Sarpal, S., & Arora, P. (2013). Analysis of different types of process models - software. *International Journal of Research in Information Technology, 1*(11), 64.

U.S. Food and Drug Administration [FDA]. (2014a). Code of federal regulations [CFR]. 21CFR § 820- quality system for medical device and equipment/software manufacturers. Retrieved June 07, 2014 from http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820&showFR=1

U.S. Food and Drug Administration [FDA]. (2014b). Code of federal regulations. 21CFR § 860 - Medical device classification procedures. Retrieved June 7, 2014, from http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=860

Unhelkar, B., & Murugesan, S. (2010). The enterprise mobile applications development framework. *IT Professional, 12*(3), 33-39.

Voas, J., Michael, J. B., & van Genuchten, M. (2012). The mobile software app takeover. *IEEE Software, 29*(4), 25-27.

Wasserman, A. I. (2010). Software engineering issues for mobile application development. In *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research,* (pp. 397-400). New York, NY: ACM.

Weintraub, A. (2013). *Healthcare.gov: Your reminder not to skimp on quality assurance.* Retrieved March 31, 2014, from http://www.entrepreneur.com/article/229859

White, C. (2013). *Idea to iPhone: The essential guide to creating your first app for the iPhone and iPad.* Chichester, West Sussex: John Wiley & Sons.

Yetisen, A. K., Martinez-Hurtado, J., da Cruz Vasconcellos, F., Simsekler, M. E., Akram, M. S., & Lowe, C. R. (2014). The regulation of mobile medical applications. *Lab on a Chip, 14*(5), 833.

Zahra, S., Khalid, A., & Javed, A. (2013). An efficient and effective new generation objective quality model for mobile applications. *International Journal of Modern Education & Computer Science, 5*(4), 36.