

# A Software Engineering Team's Perspective on the Switch to Agile Software Development

---

By: Samir Patel

Spring 2011

Engineering Management

The University of Kansas



# Table of Contents

- Acknowledgements..... 5
- Executive Summary..... 6
- Introduction ..... 8
  - Team ..... 11
- Literature Review ..... 12
  - Waterfall Software Methodology ..... 12
  - Agile Software Methodology ..... 15
  - Pros/Cons of Transitioning to Agile ..... 21
- Procedure and Methodology ..... 25
  - Survey..... 25
  - Focus Group ..... 26
  - Interpretation ..... 26
- Results ..... 28
  - Survey..... 28
  - Focus Group ..... 39
  - Interpretation ..... 40
  - Conclusion..... 41
- Suggestions for Additional Work ..... 42
- References ..... 44
- Appendix A..... 45
- Appendix B ..... 47
  - Survey Response 1 ..... 48
  - Survey Response 2 ..... 49
  - Survey Response 3 ..... 50
  - Survey Response 4 ..... 51
  - Survey Response 5 ..... 52
  - Survey Response 6 ..... 53
  - Survey Response 7 ..... 54
  - Survey Response 8 ..... 55
  - Survey Response 9 ..... 56

Survey Response 10 ..... 57  
Survey Response 11 ..... 58  
Survey Response 12 ..... 59  
Survey Response 13 ..... 60  
Survey Response 14 ..... 61

## Acknowledgements

I would like to take the opportunity to thank Linda Miller for being very supportive and helpful during the inception and creation of my project. As chair of this field project, she has provided guidance and help to make this project a success. Linda inspired me to choose a topic that aligns perfectly with my career goals.

I would also like to thank the entire EMGT faculty for provided a world-class education and for being very supportive during the duration of my education career in the program. The faculty has spent countless hours working with me to ensure I get the best education possible, and for that I am thankful.

Last but not least, I would like to thank my wife, parents, and brother for supporting me over the past three years as I spent many nights working on homework and projects. They have provided the encouragement and support to keep me going. I would not have been able to complete this program without them.

## Executive Summary

Healthcare information technology is one of the fastest growing industries in the world. There are millions of errors made in the medical paper chart at hospitals around the world, resulting in mistaken medication and death. Healthcare information technology provides a way to eliminate these errors and to allow patients access to their medical record from anywhere in the world. Corporation X is the foremost leader in the industry and has been providing solutions to hospitals and doctors' offices for the past 30+ years. Developing software that affects peoples' lives requires many processes and certifications. As a result, the software development process is instrumental in providing high quality software for Corporation X's clients.

In the past, Corporation X has employed the Waterfall Software Methodology to design and implement their software. This methodology or process has been in use since the beginning of software. There are distinct phases that need to be sequentially followed, requiring much up-front work. There are many advantages and disadvantages to this process. As the healthcare information technology industry grows and new competitors make their way to market, Corporation X has been forced to re-think this process and adapt to the ever-changing external environment.

The recent popularity of Agile Software Development has spread across the world. In the past 10 years, Agile has become the most used process for software development. In 2010, Corporation X decided to switch from Waterfall to Agile, to keep up with the industry and improve it's own processes.

The advantages of the switch to Agile have proven to be a great asset to Corporation X's engineering teams. The new process has increased code quality, increased client satisfaction, and produced more revenue for Corporation X.

There were, however, a few disadvantages such as resource pooling and team morale.. A new concept was introduced with Agile, resource pooling. This was Corporation X's own spin on resourcing, not something that is traditionally a part of Agile practices at most companies. This proved to be a disadvantage because it caused teams to lose that camaraderie and closeness that is typical of a team. Engineers had to start over with people with every project. Corporation X also put a new defect accountability process in place to help drive down the number of defects by introducing the feeling of ownership. And with this new process, engineers became more aware of the number of defects being introduced to clients. All of these factors will be discussed and explained in this study. This study will help dissect the transition from Waterfall to Agile from the software engineers' viewpoint.

## Introduction

There are many different methods for software development, such as Waterfall, Prototyping, Incremental (Agile), Spiral, and Rapid Application, each having their own benefits and disadvantages to the software development lifecycle. This paper will take an in-depth look at Waterfall and Agile.

Software prototyping is the creation of prototypes, which are incomplete versions of the software being developed. The basic principles are:

1. "Small-scale mock-ups of the system are developed following an iterative modification process until the prototype evolves from prototype to working system.
2. Attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.
3. User is involved throughout the development process, which increases the likelihood of user acceptance of the final implementation." (Beaver 2007)

Spiral development combines elements of both design and prototyping in stages, in an effort to combine advantages of top-down and bottom-up concepts. In a top-down approach, an overview of the system is formulated, specifying first-level subsystems. Each subsystem is refined in greater detail until the specification is reduced to base elements. The bottom-up approach is the piecing together of systems to give rise to grander systems. The basic principles are:

1. Break a project into smaller segments and provide more ease-of-change during the development process.



2. “Each cycle involves a progression through the same sequence of steps, for each part of the product and for each of its levels of elaboration, from an overall concept-of-operation document down to the coding of each individual program.” (Beaver 2007)
3. Each trip around the spiral traverses four basic quadrants, as shown in the figure below.
4. Begin each cycle with an identification of stakeholders and their win conditions, and end each cycle with review and commitment.



Rapid Application development involves iterative development and the construction of prototypes. The basic principles are:

1. Key objective is for fast development and delivery of a high quality system at a relatively low investment cost.
2. Attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.

These methodologies are employed by many companies around the world, but Waterfall and Agile are still the overwhelming favorites. This paper will take the Traditional Software Methodology (often referred to as Waterfall) and the Agile Software Methodology and

compare them. Both of these methodologies are used by large software companies, who produce complex enterprise solutions. The nature of the software and the company's clients dictates which approach works better.

The Waterfall method has been around since the inception of the software industry. It consists of a linear approach, which can span months. Each phase of the project must be completed before the next can begin. Agile, on the other hand, follows the same stages, but implements them in small iterative chunks. Large, complex projects are broken down into smaller, manageable pieces, which then go through each stage. There are advantages and disadvantages of both methodologies.

This paper will analyze the benefits/disadvantages of both methodologies based on Corporation X's switch from Waterfall to Agile. The analysis will take many factors into account, including management viewpoint, software engineer satisfaction/dissatisfaction, client perception, length of project, and other various indicators.

Corporation X is a healthcare information technology company that provides software for medical institutions like hospitals and doctors' offices. The nature of this software requires well designed, high quality code. This is where different software methodologies come into play. Each methodology has its benefits and disadvantages especially when applied to different software industries.

The author wanted to know the thoughts and opinions of Corporation X software engineers about the new approach to software development. Corporation X has decided to make the switch to Agile, which touts high quality, quick development, and constant feedback as its main

attributes. There have been mixed reviews with many common praises and complaints about the process. With this knowledge, the author hopes to learn how to enhance the process and help his team achieve a better future outcome with Agile.

## Team

Corporation X offers many applications/solutions for clients. As a result, teams were previously organized by domain/knowledge. Engineers worked on the same solution throughout their career, so they became experts in certain domains. With the introduction of Agile, team organization was migrated to the matrix model where engineers worked on various projects, including other solutions they were not familiar with.

With the Agile approach, all engineers are put into a “pool” from which they are selected for individual projects. The selection process is performed by management teams and is not shared outside of management. It is speculated that engineers are ranked based on past project experience and reputation, although it is not certain what other factors are used. This unique approach to team organization is a key attribute to Corporation X’s switch from Waterfall to Agile. It is a Corporation X-specific implementation of resource pooling.

Teams had been working with the new Agile Software Methodology for a little over 1.5 years when this study was conducted. Software engineers had worked on about three major projects, each lasting about six months. Software engineers were the only role that were taken into consideration for this study.

## Literature Review

As stated above, this study is about Corporation X's transition from Waterfall to Agile Software Methodology. A background on both methodologies is provided below.

### Waterfall Software Methodology

The waterfall software methodology has been around as long as software has. It was the first mainstream software development methodology used by software companies. The instinctive linear progression of projects led to this process, where each phase of a project is done in sequential order. The phases are outlined below.

“Among the traditional development approaches, the waterfall model is the oldest. It has been widely used in both large and small software projects and has been reported as a successful development approach especially for large and complex engineering projects.” (Ming, Verner, Liming, Babar, 2004) It is no surprise Corporation X started off using this methodology because of the complexity of its software. “The waterfall model divides the software development lifecycle into five distinct and linear stages. Some stages can be overlapped, but it generally follows a linear flow. The five stages are:” (Ming, Verner, Liming, Babar, 2004)

1. Requirements analysis and definition
2. System and software design
3. Implementation and unit testing
4. Integration and system testing
5. Operation and maintenance

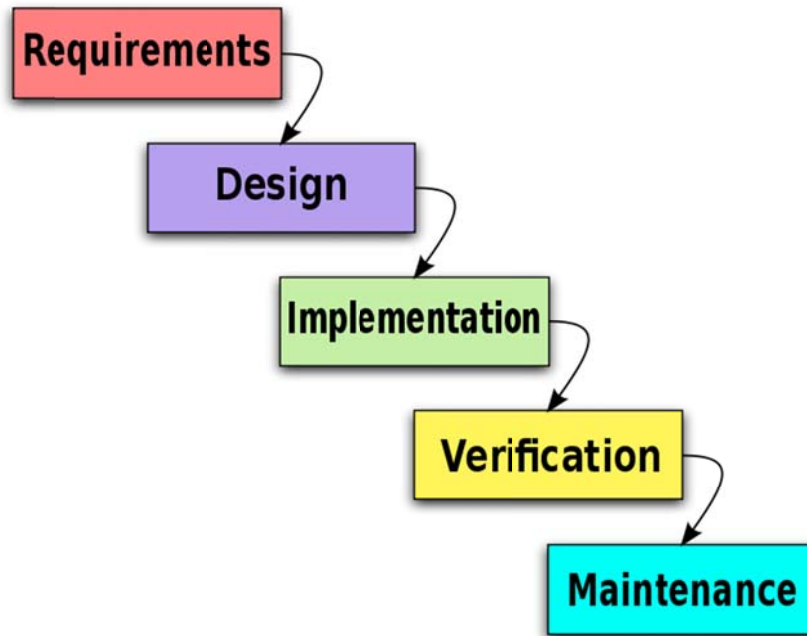


Figure 1: The progression of the five stages of Waterfall Software Development.

“Each stage produces well-defined deliverables. Since the deliverables of one activity are input for a subsequent activity, no subsequent stage can begin until the predecessor stage finishes and all of its deliverables are signed off as satisfactory.” (Ming, Verner, Liming, Babar, 2004)

Each stage can take anywhere from one day to a couple months based on the complexity of the project. This goes to show how it is difficult for companies to quickly adapt to changes and shifts in the market or client demand. Many times clients have to react to their external environment due to competitors or a change in technology. These changes in environment translate into changes in the software requirements. A client may also need to update software functionality because of changes in their internal environment. A shortage of funding may reduce the scope of the project. A shift in the client’s mission or vision can also cause software projects to change instantaneously.

The linear stages require one stage to be completed before the other and all stages are only done once. This forces software teams to spend quite a bit of time to finalize requirements and design before any coding can begin. Once coding has started changes in requirements delay the project dramatically because the entire process needs to start over. In complex projects, it is difficult to have all requirements finalized before the project starts, so it is fairly common to encounter this situation. "Some of the drawbacks of the waterfall software methodology are the inflexibility in the face of changing requirements, and a highly ceremonious process irrespective of the nature and size of the project." (Ming, Verner, Liming, Babar, 2004)

Many software engineering teams swear by the Waterfall methodology and its concrete process. Some argue that well-designed software needs to be fully planned before any development starts, which is the opposite of the iterative development concepts of Agile. "Waterfall is a heavyweight methodology, which promotes up-front overall planning of the roles to be played, activities to be performed, and artifacts to be produced. Proponents of these methodologies argue that planning leads to lower overall cost, timely product delivery, and better software quality." (Meso, Jain, 2011)

There are many advantages and disadvantages of implementing the Waterfall software methodology. Certain factors, such as technology, limit companies to use certain methodologies. Companies that do not have automated testing will have a difficult time using Agile because of the short repetitive testing iterations. This is just one example of how many software firms are forced to use Waterfall.

This process has been successful because it is easy for companies to follow this natural progression of stages. Most software is complex and it takes a large amount of time to work through what the software does (requirements) and how it will be implemented (technical design). These two tasks must be completed before coding can begin, especially for detailed software programs. The traditional viewpoint was to gather all requirements, so that the engineering team was aware of what needs to be done before starting. This allowed the software architects to consider all scenarios when designing the software, which reduced the amount of re-work in later stages of the project. This kept programmers happy because they had a detailed plan of how the software will be laid out. This also reduced the amount of delays for the same reason that everything was already thought through. As companies evolve and software becomes more commonplace, this philosophy of linear progression will expose its flaws.

## **Agile Software Methodology**

The Agile Software Methodology has a simple motto, referred to as the Manifesto. The principles behind this methodology are based on this manifesto:

“We are uncovering better ways of developing  
Software by doing it and helping others do it.  
Through this work we have come to value:”

<u>Agile Software Development</u>	<u>Waterfall Software Development</u>
Individuals and interactions instead of...	Process and tools
Working software instead of...	Comprehensive documentation
Customer collaboration instead of...	Contract negotiation
Responding to change instead of...	Following a plan

(Agile Manifesto, 2001)

That is, while there is value in the items on the right, we value the items on the left more. This manifesto gives a good idea of the basic principles of the Agile Software Methodology. The twelve principles of Agile are:

1. "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.



8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity, the art of maximizing the amount of work not done, is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. Disagree.” (Agile Manifesto, 2001)

These 12 principles guide the behavior of teams following the agile software methodology, which differ quite a bit from Waterfall. Principle 2 explains the welcoming of changing requirements. This conflicts with Waterfall’s inability to adapt to change. Principle 3 explains how the project is broken up into small iterations, each with deliverables that can be taken to the client. The Waterfall approach doesn’t foster the ability to deliver intermediate results throughout the lifecycle. A working product will most often only be available at the end of the entire project. Principle 10 talks about keeping the code simple. Breaking a complex project into smaller, manageable pieces helps simplify the intricacy. The Waterfall methodology tries to tackle the entire project at once, adding to the complex nature of the project. Not only is the algorithm detailed and difficult to implement, Waterfall makes the process of developing that software more cumbersome and stressful. Finally, principle 11 hints to the idea of self-organizing teams. This is one of the strongest aspects of Agile. The idea that people who work well together will gravitate towards each other and form teams that will succeed with the task at hand. This is part of resource pooling, in other words how engineers are pooled together for projects. With Agile, these teams can change in between any iteration, promoting the idea

“make it work.” (Miller, Larson, 2005) Waterfall, on the other hand, maintains its teams throughout the life of the project. This doesn’t encourage the self-formation of cohesive teams. The burden is put on the project manager, giving him/her another thing to stress about. These principles clearly emphasize the importance of iterative development, along with continuous feedback and other important factors, which differ tremendously from Waterfall.

Many companies have recently switched to Agile and have agreed the benefits far outweigh any disadvantages. “Agile software development has gained tremendous acceptance since the late 90s because it accommodates volatile requirements, focuses on collaboration between developers and customers, and supports early product delivery. Two of the most significant characteristics of the agile approach are 1) it can handle unstable requirements throughout the development lifecycle and 2) it delivers products in shorter timeframes and under budget constraints when compared with traditional development methods.” (Ming, Verner, Liming, Babar, 2004) “The agile approach turns the traditional software process sideways. Based on short releases, agile methods go through all development stages a little at a time, throughout their software development life cycle.” (Ming, Verner, Liming, Babar, 2004) This agility allows companies to shift resources and priorities as the market demands. Arguably, the most beneficial attribute is the guarantee of always having a deliverable piece of software. Each iteration results in a working product so that if scope is reduced, there is something to show for the work already put into developing the software.

“Lightweight or agile methodologies put extreme emphasis on delivering working code or product while downplaying the importance of formal processes and comprehensive

documentation. Proponents of these methodologies argue that by putting more emphasis on 'actual working code', software development processes can adapt and react promptly to changes and demands imposed by their volatile development environment. (Meso, Jain, 2011)" This proves to be an important advantage for software companies because they are capable of changing functionality of the software whenever the client decides. A company can use this strategy as an advantage over other companies.

"Literature on agile software development suggests that for a complex systems development project to be successful it should be implemented in small steps, each with a clear measure of achievement and with an option of rolling back to a previous successful step upon failure." (Meso, Jain, 2011) This principle allows companies to always have a deliverable ready for their clients, even if there is an issue or failure in the later stages of the project.

Another major benefit of Agile is the continuous feedback from clients. "Software development by its nature involves interactions among stakeholders, development tools, software components in the solution being developed, and interactions of stakeholders with these tools, solution, or both. These interactions are message exchanges occurring between entities related or interconnected in some fashion to each other." (Meso, Jain, 2011) "When a development team establishes transforming feedback loops across a boundary that divides the team from users, other development agents, and stakeholders, a complex adaptive development environment is created." (Meso, Jain, 2011) This continuous feedback loop results in a software product that exactly matches the needs of the clients. As the software is developed, clients may realize changes in functionality or appearance and these changes can be addressed with the

iterative approach of Agile. The figure below illustrates the progression of an agile software project.

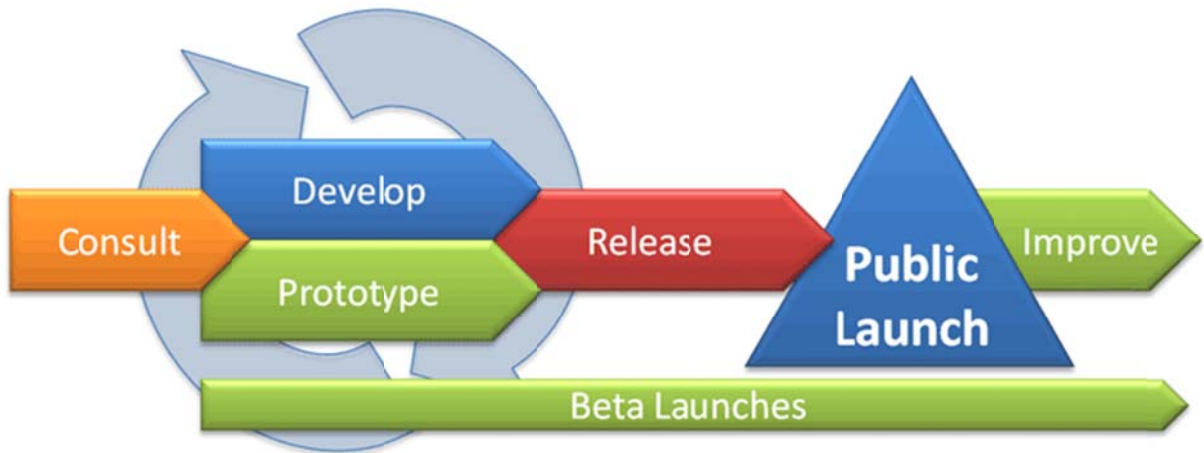


Figure 2: The progression of the stages of Agile Software Development.

The main trait of Agile is the iterative approach. Agile contains the same stages as the Waterfall method, but the stages occur in a much smaller timeframe. The full lifecycle is typically performed between 2 weeks and 1 month. Large projects are broken into smaller iterations, so that engineers can focus on each piece, producing higher quality software. This results in functional software at the end of each iteration.

Agile has many benefits, as stated above. The most important benefit is the ability to adapt to changes. "One of the major attributes of agile software development methodologies is that they embrace changes in requirements from various sources, even in later development phases. This underscores an emphasis of agile methods on willingness to embrace emergent requirements and unanticipated changes regardless of when they occur in a project's lifetime. It is important that any changes to project requirements are handled proactively by making

necessary adjustments to the development process and the development team.” (Meso, Jain, 2011)

## **Pros/Cons of Transitioning to Agile**

The transition from Waterfall to Agile can be a tough process, requiring much research and planning. As this paper has stated, Agile brings many benefits to a software company. Along with those benefits, come disadvantages of switching. This section will go over the pros and cons of switching from waterfall to agile.

The typical lifecycle of a software project, using Waterfall, can range between two months to two years, depending on the scope. This large timeframe poses a problem for development teams because as the beginning phases are completed, they are difficult to revisit as the project moves further and further along. Requirements are hashed out at the beginning of a project. If changes are needed a couple months into the project, it is difficult to re-work everything that has been done so far. This is where the benefits of Agile come in to play. The typical software project, using Agile, can have a similar range as Waterfall. However, the difference is that the project is broken up into smaller pieces, so the project lifecycle is repeated in small increments. “This approach allows the project to adapt to any changes in demand.” (Lindstrom, Jefferies, 2004) This is a huge benefit over Waterfall.

Another benefit of transitioning to Agile is the increase in team morale. As a software engineer, you have a love-hate relationship with the process. It can either foster the engineers’ creativeness or slow everything down. Software companies should know that engineers chose this profession because they want to create software. They want to spend their hours coding,

not dealing with process. As irritating as processes can be, it is mandatory to have one. One of the biggest process annoyances for a software engineer is the lack of planning and having to redo work. With Waterfall, a one-year long project's requirements are finalized at the beginning of the project, so changes are difficult to make and cause much rework. With Agile, complex projects are broken up into smaller chunks so that each piece of functionality can be well planned and architected. This allows the engineer to code small chunks while keeping in mind the overall goal of the project, which keeps the amount of rework to a minimum. Another nuisance for engineers is waiting for requirements or technical designs to be completed before coding can start. Agile's smaller iterations are good because initial phases, like requirements and technical designs, can be completed in shorter amounts of time. In the Waterfall approach, engineers can sometimes wait as long as a month to get requirements and design finalized. This is a huge concern from a management perspective because engineers are waiting around to get started on projects. With Agile, that is no longer a concern.

Wipro is a software consulting firm in India. A study done by Wipro employees says "moving from waterfall to agile cannot be done overnight or in a single step. It takes time for people to unlearn old traditional [waterfall] practices and move towards agile. Coaches need to be patient, positive, and persistent while bringing in this change. Agile practices call for a change in mindset." (Sureshchandra, Shrinivasavadhani, 2008) There are many hurdles to jump when transitioning between the two methodologies, but it is well worth it. "Long-term benefits include increased project awareness for new team members, restored confidence in the original project concept, and the inclusion of clear and present business critical design requirements that were not evident at the project inception and thus fell out of scope. While

waterfall projects fail for many reasons, there is a direct correlation between failed projects and overall project timescale. Part of the reason for this is that long projects often see a fair level of staff turnover. New team members may not understand the original project or may have been fully integrated when coming onboard. After a certain period of time, even the original team members have difficulty maintaining a clear view of project goals, no matter how well they were articulated at the onset of the project. Switching to agile methods gives organizations the opportunity to review project goals, analyze past performance, and refresh the project team's understanding." (Heath, 2007)

There are also some short term benefits. "As costs rise and no deliverables are produced, failing projects often fall under the critical eye of top-level management. A switch to agile allows a project to start producing recognizable results in the short term. This will help to raise the falling opinion and remind the organization of the initial perceived value of the project. A switch to agile is especially beneficial when a project is struggling to pass the test and review stage. The agile method is especially suitable in this situation as it inherently promotes quick turnaround, fast response, and instant solutions." (Heath, 2007)

Companies have an important decision to make when it comes to implementing a software development methodology. There are many factors that come into play when selecting the most appropriate process, such as technology, project size, company size, and funding. Established companies tend to stay with the Waterfall Software Methodology because of the experienced engineers who have been using that method throughout their careers. Agile is a

relatively new concept which is common among newer companies with a younger group of software engineers.

The ability to switch from one methodology to another is dependent on the willingness of management and engineers to take the extra time and effort to put new processes in place that will completely change the way software is developed in their organization. The information about both methodologies, given in the above sections, provides a good understanding of what the stages are for each approach. Switching to Agile requires the ability to perform the same tasks, but in smaller iterative chunks of time. The benefits are not only internal to the company, but external also. Clients have a better perception of the company's software quality and ability to listen to their requests. The continuous feedback loop provides a unique mechanism to involve the client in all stages of the development, resulting in a software product that will surpass all initial expectations.



## Procedure and Methodology

### Survey

For this study, a survey was used to gather data, from software engineers. The survey questions were developed based on research and the positive and negative “chatter” heard throughout the transition to Agile. The switch to Agile was a huge task to undertake, due to the enormous impact it would have on the company and due to the size of the development organization. One of the biggest challenges was getting management and the engineers on board with this new philosophy. To convince software engineers, a pilot program was initiated as a feasibility study. The results showed the positive impacts of Agile, such as quick adaptability to change and the perception of less work because of how it was spread out over iterations.

During meetings and in the break room, a lot of opinions were voiced on how the transition is going. Those opinions are what helped derive the questions for the survey. Many engineers think there are many benefits, but there are quite a few drawbacks. Comments like “there is a guy working on this project who knows nothing about this domain”, show the dissatisfaction with the way resourcing is handled. Comments like “breaking up this project into smaller chunks really helps ease stress”, show the benefit of breaking large complex projects into manageable iterations. These are the types of “chatter” that shaped the survey.

Agile Software Development is now being taught in universities across the country, so more and more college graduates are familiar to, if not used to, that approach. Corporation X’s young engineering population helped ease the transition. The older software engineers were more hesitant to change because they know only one way to develop software, which is with

Waterfall. This diversity in the engineering organization is a great reason to study the effects of Agile.

Most software engineers had worked on at least one full iteration using the new process, so everyone had substantial time to develop their take on how the new methodology was working prior to the survey. The survey was given out to 20 people within the development organization, primarily teams the author is in constant contact with. 14 of those 20 responded. The results were cross-tabbed in Survey Monkey.

The survey is attached in Appendix A.

### **Focus Group**

Upon completion of the survey, a focus group was conducted with 14 people for 30 minutes. The purpose was to create an open forum for engineers to voice their opinion and expand on their responses. More focus was put on the 4 engineers that had a negative perception on the transition to Agile.

### **Interpretation**

Another interesting interpretation of the study was that the real problems with the switch to Agile were not due to Agile itself, but due to other changes made by Corporation X that were specific to that company. Resource pooling and the defect accountability program were the two most complained about attributes of Agile, and those are the only two attributes that are not part of the Agile Software Methodology.

It was very interesting to see that the four software engineers that had negative views on the transition to Agile were older engineers. They had been used to Waterfall and weren't very open to change.

## Results

### Survey

The survey was given to twenty software engineers. Fourteen out of those twenty engineers responded. Those results are attached to this paper in Appendix B. The software engineers that took the survey were part of the author's engineering team. The results of the survey were manually entered into SurveyMonkey.com, from which data was analyzed and put into charts. The results of each of the questions are included below with a brief summary of its significance.

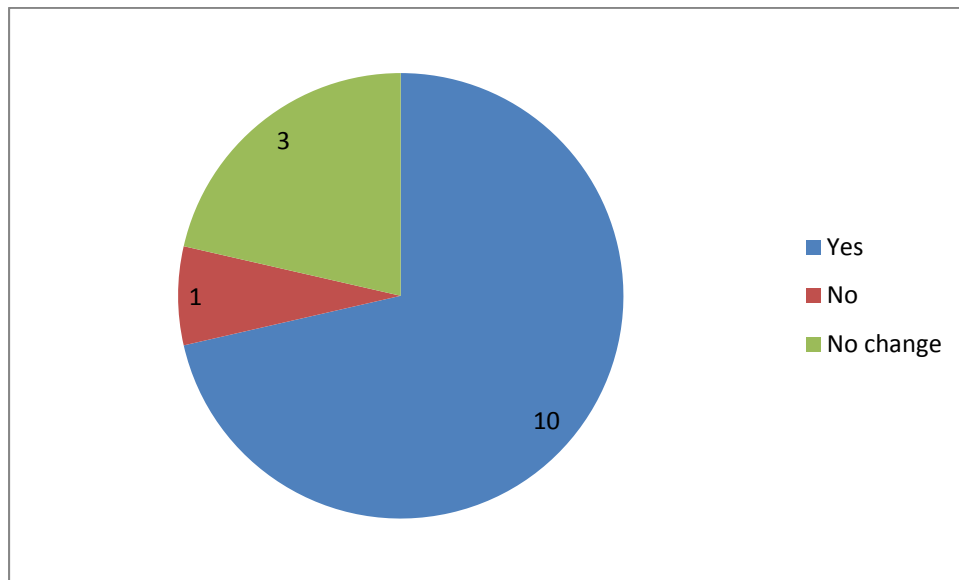


Figure 1: Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?

Figure 1 shows how software engineers feel about the progression of software projects using Agile. 71.4% of responders think Agile has had an overall positive effect on projects. This shows that even though there may be downfalls to Agile, the general opinion is positive. The whole goal of making the switch was to streamline and simplify the development process by applying

the new principles of Agile. Upon further discussion with the group of engineers, it was revealed that the main contributor to the positive outlook was the Agile principle of breaking large projects into smaller iterations. This played a significant role in simplifying projects and reducing the stress level of the engineers. With Waterfall, it was much more difficult to get a project to completion, because all the work was done in a single iteration, resulting in procrastination and a big push at the end. A side-effect of the rush at the end of the development cycle was poor code quality. In the healthcare information technology industry, software quality is of the utmost importance and can have a huge impact on the bottom line. The results to this question show that Corporation X had the right idea when making the transition to Agile.

When designing this survey, the author kept the three major players in mind. The players are the team, the company, and the clients. The success of the transition to Agile will be gauged by the level of satisfaction and positive effects on all three players. The following three questions (Figures 2, 3, and 4) were asked to the engineers to get their take on how each major player was affected.

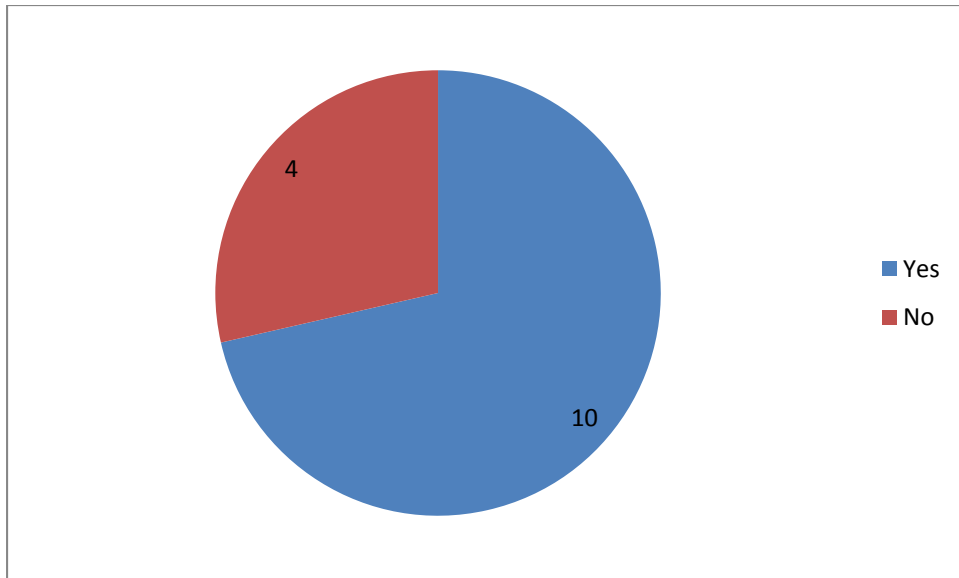


Figure 2: Do you think the switch from Waterfall to Agile was beneficial to the team?

Figure 2 shows how the software engineers felt about the effects of Agile in regards to their team. Engineers were asked if they thought the switch from Waterfall to Agile was beneficial to their team. 71.4% of participants concluded that their team benefited from the new methodology. Many of the principles of Agile played into the response for this question, like splitting up large projects into smaller, more manageable pieces and having deliverables ready at the end of each iteration. Software engineers like to reuse as much code as possible, so when complicated functionality is broken up into smaller pieces, there are more reusable components that can be used in future projects. This helps keep code quality high and also saves in development time, so that engineers can use that saved time and focus on other strategic projects.

Team morale is also very important in any industry. With software development companies, it is very important to have a team culture that fosters creativity and a low-stress environment. When engineers work on software that has the potential to save or harm a person's life, there

is a huge burden put on their shoulders. This stress needs to be reduced, so that development teams are not bogged down. The above mentioned attributes of Agile do exactly that and the survey respondents agree. The few software engineers that disagreed all disagreed on the same questions. It was interesting to note that they were all older than the rest of the engineers.

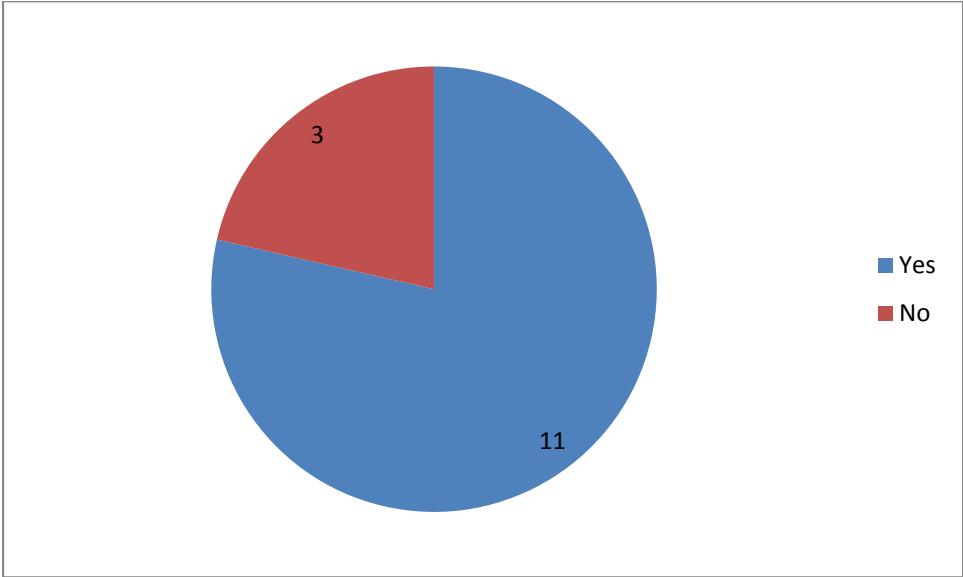


Figure 3: Do you think the switch from Waterfall to Agile was beneficial to the company?

Figure 3 asks engineers if they think the switch from Waterfall to Agile was beneficial to the company. 78.6% of participants replied that Agile is more beneficial. The purpose of this question was to find out what engineers thought about the effect of Agile on the company as a whole. With this transition in methodologies, there are many areas of the company that were affected. Processes had to be changed which had to remain compliant with health & government regulations, in order to stay certified. All of our processes are audited by regulatory agencies (HIPAA, ISO, FDA, etc.). Many teams work to make sure we stay certified, not just development teams. As an engineer, it is sometimes difficult to see what other parts of

the company are affected by a switch in methodology like this. This is a potential area of further study, to see how the new methodology affected other parts of the company.

The results show that most engineers thought the switch to Agile was a good thing. This is because projects are well-documented and have deliverables early in the stages of development. That allows teams all throughout Corporation X to complete their part of delivering Corporation X's mission. Engineers only develop the code. There is much documentation and support that goes along with it, which makes up a large part of Corporation X. With Agile, those teams are allowed more time to do their part before the end product is shipped to clients. This results in a better quality product. It's a domino effect that keeps going to the client and then back to Corporation X in the form of revenue, which is the most important reason for staying in business.

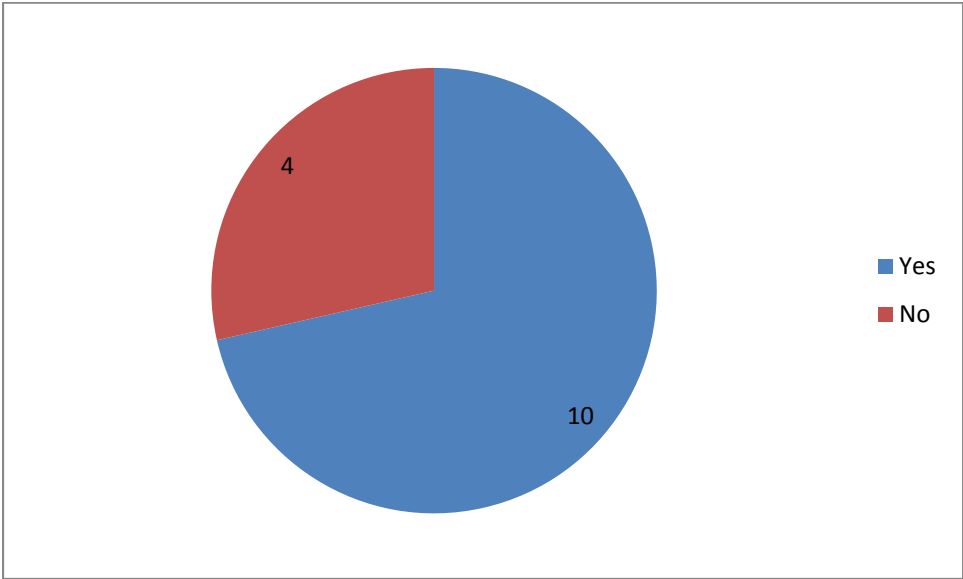


Figure 4: Do you think the switch from Waterfall to Agile was beneficial to the clients?



Figure 4 shows the results of asking the engineers if they thought the switch to Agile was beneficial to the clients. 71.4% of participants responded yes. From the software engineer's perspective, it is sometimes difficult to comprehend what kind of affect the software development process has on clients. It often feels like an internal process that has no direct effect on the client, but that is not true. Engineers learn that with Agile the clients are involved from the beginning. Projects are prioritized by client demand and are defined by the clients. With Agile, all projects are broken into shorter iterations, each with a deliverable. These deliverables are taken to the client, to ensure the project is on the right track. This creates a transparent communication between engineering teams and the client, where engineers get to witness firsthand what clients think about the software. After each iteration, requirements can be changed by the client, so that the software is tailored exactly to their need. This creates an incredible advantage for Corporation X over other companies that do not employ this methodology.

The software engineers notice the increase in client satisfaction by the increase in interaction with them. They get to see how excited clients get when they get to design the software, instead of being forced to use something that was devised by engineers who have limited functional knowledge of how the software is used in a clinical setting. The increase in number of requested projects by clients has also increased, given engineers another way to see the satisfaction level. This is an extremely huge benefit of Agile that Corporation X engineers have noticed.

The next question asked to engineers was aimed at getting to know on an individual level what they thought about each of the main advantages of Agile. Based on the literature review, there are many advantages to Agile, including meeting customer needs, improving software quality, faster delivery time, lower development costs, increase in reusable code, and increased flexibility in development. The software engineers were asked to determine if each of these six attributes increased on a scale of:

Strongly Agree      Agree      No Difference      Disagree      Strongly Disagree

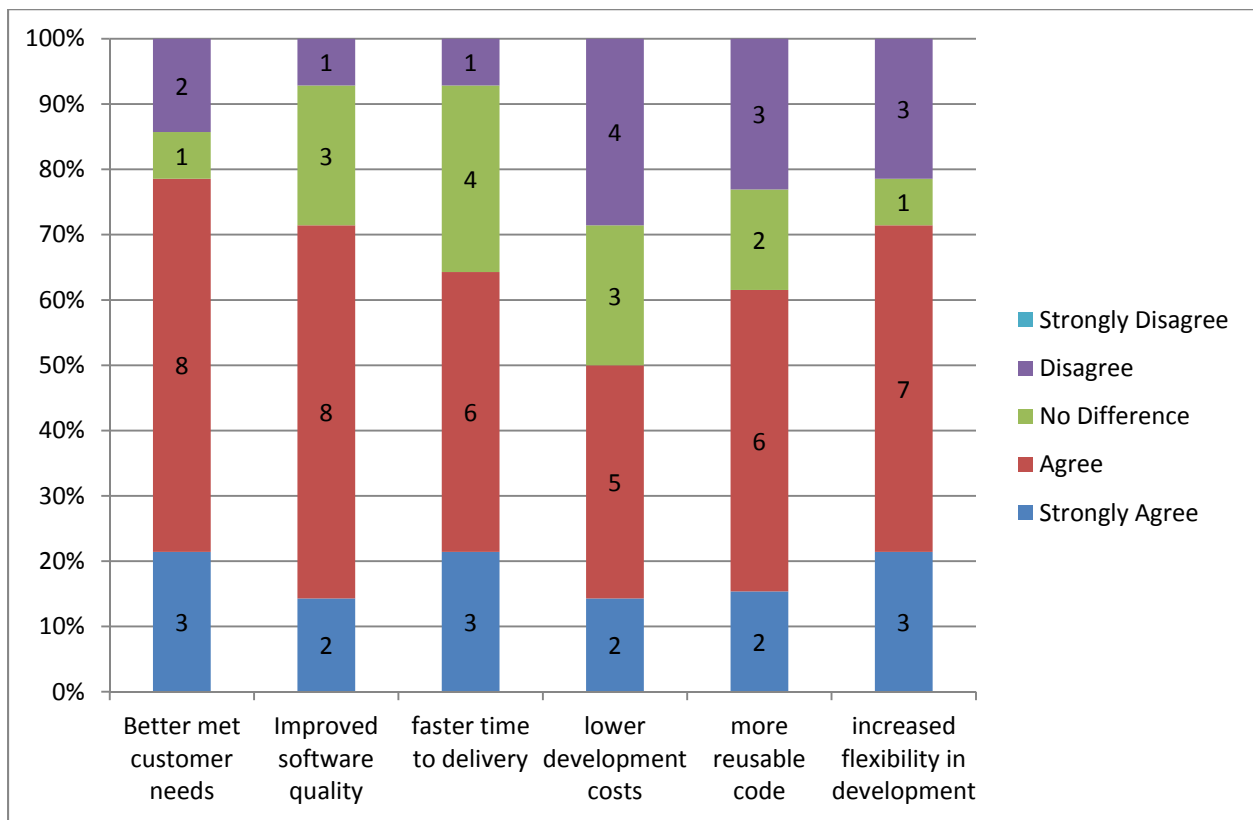


Figure 5: The switch to Agile has...

Figure 5 shows the number of participants that ranked each attribute at each level. The x-axis shows each attribute and the legend shows the different ranks. Based on data, it is apparent

that most engineers agree that the six major attributes of agile has really applied to Corporation X's transition. It was interesting that not a single engineer strongly disagreed with any of the attributes. There were a few that disagreed, but the general consensus was positive. The results to this question correlate with the how the engineering team felt about Agile's effect on the team, the company, and the client. Each of these attributes relates to those three major players in the transition to Agile. It is important to note that the two strongest attributes were "better met customer needs" and "faster time to delivery". These two attributes are incredibly important to a company's success because it directly affects the clients. The time to deliver a finished product is crucial to the bottom line. The more time a team spends on developing a product, the less money the company will make on it and the less happy the client will be. Agile enables development teams to deliver products in a timely manner, thus increasing customer satisfaction.

Figure 5 also shows that the attribute that engineers disagreed with most was the decrease in development cost. This is mainly because cost is not apparent to engineers, so it is hard to gauge if the transition was really successful in this area, without interviewing the correct software engineers. Engineering managers have the metrics on cost, in terms of FTE (full-time engineer) hours and dollars, which are kept at the management level. Another reason engineers may think of this attribute as negative is because of the way resource pooling is done. With Waterfall, engineers belonged to a team that would work on the same concept or domain. They would be the domain experts, so any project related to their domain would be handled by the same group of engineers. A new concept was introduced with Agile, resource pooling. This was Corporation X's own spin on resourcing, not something that is traditionally a

part of Agile practices at most companies. Corporation X decided to put all engineers into a single pool. When a project was accepted, managers would select resources/engineers from the pool to go work on these projects. It wasn't guaranteed that that engineer A would be experienced in domain B, whereas with Waterfall, engineers were always working with the same code. This can be good and bad. Many engineers like the variety, but also lose that sense of ownership when they are moved around every project. Software quality is also affected because engineers who are familiar with the subject matter are working on something completely different that they have no prior experience working with. This plays a large role into why developers may think that the development costs are actually increasing because of the amount of time needed to get up to speed on a new domain.

The survey included a specific question around software quality because that is one of the major benefits of Agile. Corporation X has had problems with software quality in the past and engineers have been made well aware of it. As the executive management team has made a push to increase quality, Agile has incorporated processes that helped in this area.

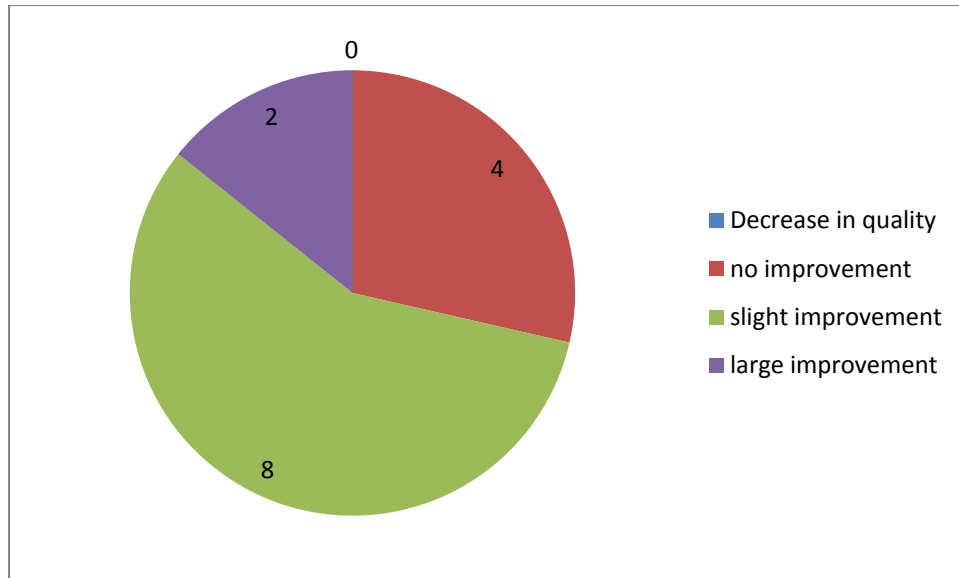


Figure 6: How would you rate the improvement (# of defects) in software quality due to Agile practices?

Figure 6 shows the responses given by software engineers about the improvement in software quality due to the Agile processes. Software quality is generally based on number of defects. Corporation X keeps track of the number of defects generated by each engineer, so that engineers feel a sense of accountability for what they write. The idea was without a way to link defects to individual engineers, there is a lack of ownership. This new defect accountability process was put into place to help drive down the number of defects by introducing the feeling of ownership. And with this new process, engineers became more aware of the number of defects being introduced.

Agile has many processes in place to help increase software quality. The break-up of large projects into smaller iterations helps reduce the complexity of code that needs to be tackled at once. The complex logic is split up, so that it can be well thought out and designed to eliminate defects. The code is also reviewed by peers in each iteration, so by the time the project is complete, the code has been reviewed multiple times by multiple engineers, reducing the

number of bugs that are introduced to the client. This is completely opposite of how Waterfall methodologies deal with projects. Each project is done in one iteration and the complex algorithms need to be designed and written in one pass-through. Once the code is complete, a group of engineers reviews it once and the code is shipped. The presence of repetitive reviews is an important benefit of Agile, as it tremendously increases the code quality, leading to happier clients.

The survey also asked software engineers what they thought about the clients' satisfaction levels after Corporation X's switch from Waterfall to Agile. Like stated above, it was difficult for engineers to know what clients thought about the software because they were rarely in contact with them. There is a support team that deals directly with clients, so much of the feedback is shielded from the development organization.

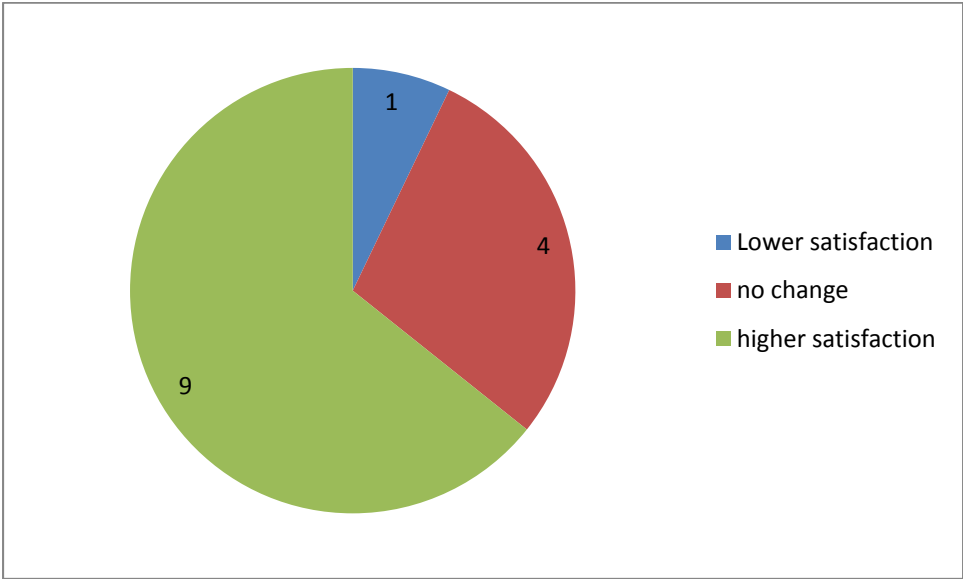


Figure 7: How would you rate Corporation X clients' satisfaction after the switch from Waterfall to Agile?

Figure 7 asks engineers how they would rate Corporation X clients' satisfaction after the switch from Waterfall to Agile. This is a very important question for engineers because they need to

know what clients think and want. Software engineers need to have the ability to design software according to client satisfaction. With Agile, engineers are given a feedback loop that is very important in relaying disappointments, successes, expected features, etc.

Software engineers were also asked what the biggest benefit/downfall was of Agile and biggest benefit of Waterfall. Most engineers thought the biggest benefit of Agile was client satisfaction and better software. The biggest downfall of Agile was resource pooling. The engineers thought the biggest benefit of Waterfall was how resourcing was done. Teams were retained and engineers were assigned to projects on which they had domain knowledge. These responses directly correlate with the graphs shown in the results section above.

## **Focus Group**

It was interesting to see that many engineers did not think there was a change in client satisfaction. When asked about it, the most frequent reasoning was because they knew more could be done to help clients, but time was not permitted or the expertise was not there. With the way engineers are assigned to projects, it is sometimes hard to fully understand a certain domain or area of expertise when you are thrown into a project without any time to research. This is the case with many projects at Corporation X. As engineers work on projects, client feedback comes in asking for a special functionality that will really make the software perfect. But the engineering team can't deliver it because of a lack of expertise. If resources stayed in the same domain, like in the Waterfall model, this would be possible. Teams also lose valuable time at the beginning of a project, as engineers are trying to gain as much domain knowledge as possible. This lost time translates into lost opportunities to fulfill clients' wishes.

## Interpretation

The survey and focus group results were interpreted by the author to determine a couple issues non-related to Agile itself. The issue with resource pooling has been brought to the attention of many executives and a new process is being piloted. Resourcing was proven to be one of the most challenging parts of implementing Agile. Teams have traditionally been together for years and now, all of a sudden, they are being pulled off to work on project with other people they have never worked before. Engineers lose a sense of team and are also forced to “break the ice” repeatedly with new engineers, failing to maintain the comfort level they used to have. This can also be a good thing because new teams can create a new innovative dynamic that produces extraordinary code. There have been many cases that support this, where random engineers were placed on a team and they worked so well together that their projects exceeded all levels of measurement and clients were absolutely thrilled with the outcome.

From the focus group, the author was able to extract the second issue. Corporation X’s defect accountability initiative was the second issue that caused quite a bit of discomfort for engineers. This process created some pressure to ensure code quality was higher, but it also introduced the “blame game.” This inadvertently put engineers against other engineers when defending the root cause of a software defect. Team morale was lowered by this because engineers felt like they were always on the defensive. Not only were they defensive, but they were dealing with team members they had never worked with, making it more uncomfortable and stressful.



## Conclusion

There is still work to be done to perfect the Agile Software Methodology. As Corporation X engineers work with management to provide valuable feedback, projects will run smoother and clients will be more pleased. Many of the processes used with Waterfall can be carried over to create a sort of hybrid Agile/Waterfall methodology. Waterfall has been employed as the methodology of choice for the past 30 years, proving to be quite successful, but as times change and as technology life cycles shrink, the need is here to update processes. It is vital to Corporation X's success that monitoring of and feedback from all teams across the company continue to happen, so that the process can be perfected.

## Suggestions for Additional Work

Due to limitations, there are a few ways to further this study that were not able to be completed. The first would be to get the opinions from upper-management. What do they think about the switch to Agile? Has it been a positive experience for their organizations? This will give another perspective to how the transition has gone. There are many metrics that are used by management teams that are not available to the software engineers, so there may be a difference in opinion on many of the attributes of Agile and how successful it really is.

Another area of additional work would be to expand the survey to groups outside the development organization. Much of the work around producing software and getting it out the door occurs separate from the actual engineering teams. There are business analysts who process client data to determine if the company is meeting metrics and deadlines. There are documentation developers that come up with all the reference manuals and documentation that go along with a complex software solution. All of these other groups are affected by the Agile process. When software is developed in iterations, a deliverable is produced every couple weeks. How does this affect other teams? Do the documentation developers have to create reference manuals each iteration? These are all questions that can be addressed to get a more comprehensive understanding of how the transition went.

One last area of additional work would be to get executive management teams involved. Really dig in there and get actual numbers regarding software quality and revenue. See if the company is really making more money because of the switch to Agile. Interview different executives and get their take on how clients are accepting the new processes and how other internal groups

are handling the change. It might also be a good idea to speak directly with Corporation X's clients to get their opinion on how Corporation X is doing, now that it employs a different software development methodology. It would prove to be very valuable to Corporation X to know what exactly the clients don't like, so that the process can be updated.

All of these are options on how to further the study. There is also the option of researching other viable methodologies. The introduction stated methods like prototyping, rapid application development, and spiral. Case studies can be performed on these methods and compared to Corporation X's Agile approach. There are many options when it comes to software development methodologies. Each can be researched to find the best fit.

## References

- "Agile Manifesto". 2001. <http://agilemanifesto.org/>. [Accessed February 2011].
- Beavers, P.A.. 2007. "Managing a Large Agile Software Engineering Organization". *Agile*: 296-303.
- Heath, Anthony. 2007. "Why Switch a Failing Waterfall Project Over to Agile?". Scrum Alliance. <http://www.scrumalliance.org/articles/73-why-switch-a-failing-waterfall-project-over-to-agile>. [Accessed February 2011].
- Lindstrom , Lowell; Jeffries, Ron. 2004. "Extreme Programming and Agile Software Development Methodologies". *Information Systems Management* 21, no. 3, (July 1): 41-52. <http://www.proquest.com/> (accessed March 1, 2011).
- Meso, Peter; Jain, Radhika; 2011. "Agile Software Development: Adaptive Systems Principles and Best Practices". *Information Systems Management*, no. 3: 19-30.
- Miller, K.W.; Larson, D.K.. 2005. "Agile Software Development: Human Values and Culture". *Technology and Society Magazine, IEEE*: 36-42.
- Ming, Huo; Verner, J.; Liming, Zhu; Babar, M.A.. 2004. "Software Quality and Agile Methods". *Computer Software and Applications Conference*, vol. 1: 520-525.
- Roche, G.; Vasquez-McCall, B.. 2009. "The Amazing Team Race: A Team Based Agile Adoption". *Agile*: 141-146.
- Sureshchandra, K; Shrinivasavadhani, J. 2008. "Moving From Waterfall to Agile". *Agile*: 97-101.

## Appendix A

The survey used for this study is below:

1. *Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?*
  - a. Yes
  - b. No
  - c. No Change
  
2. *Do you think the switch from Waterfall to Agile was beneficial to the team?*
  - a. Yes
  - b. No
  
3. *Do you think the switch from Waterfall to Agile was beneficial to the company?*
  - a. Yes
  - b. No
  
4. *Do you think the switch from Waterfall to Agile was beneficial to the clients?*
  - a. Yes
  - b. No
  
5. *The switch to Agile has... (rank each one of the below from: Strongly Agree, Agree, No Difference, Disagree, Strongly Disagree)*
  - a. Better met customer needs.
  - b. Improved software quality.
  - c. Faster time to delivery.
  - d. Lower development costs.
  - e. More reusable code.
  - f. Increased flexibility in development.
  
6. *How would you rate the improvement (# of defects) in software quality due to Agile practices?*
  - a. Decrease in quality
  - b. No improvement
  - c. Slight improvement
  - d. Large improvement

7. *How would you rate Corporation X clients' satisfaction after the switch from Waterfall to Agile?*
  - a. Lower satisfaction
  - b. No change
  - c. Higher satisfaction
  
8. *What is the biggest benefit of Agile?*
  
9. *What is the biggest downfall of Agile?*
  
10. *What was the biggest benefit of Waterfall?*

## Appendix B

The survey results are included below:

## Survey Response 1

**1. Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?**

Yes

**2. Do you think the switch from Waterfall to Agile was beneficial to the team?**

Yes

**3. Do you think the switch from Waterfall to Agile was beneficial to the company?**

Yes

**4. Do you think the switch from Waterfall to Agile was beneficial to the clients?**

Yes

**5. The switch to Agile has...**

	Strongly Agree (1)	Agree (2)	No Difference (3)	Disagree (4)	Strongly Disagree (5)
Better met customer needs.		X			
Improved software quality.		X			
Faster time to delivery.	X				
Lower development costs.			X		
More reusable code.		X			
Increased flexibility in development.	X				

**6. How would you rate the improvement (# of defects) in software quality due to Agile practices?**

Slight improvement.

**7. How would you rate Cerner clients' satisfaction after the switch from Waterfall to Agile?**

Higher Satisfaction.

**8. What is the biggest benefit of Agile?**

Large projects are broken up into smaller iterations.

**9. What is the biggest downfall of Agile?**

The resource pool is very ineffective.

**10. What is the biggest benefit of Waterfall?**

The entire project is planned out ahead of time.



## Survey Response 2

**1. Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?**

Yes

**2. Do you think the switch from Waterfall to Agile was beneficial to the team?**

Yes

**3. Do you think the switch from Waterfall to Agile was beneficial to the company?**

Yes

**4. Do you think the switch from Waterfall to Agile was beneficial to the clients?**

Yes

**5. The switch to Agile has...**

	Strongly Agree (1)	Agree (2)	No Difference (3)	Disagree (4)	Strongly Disagree (5)
Better met customer needs.	X				
Improved software quality.	X				
Faster time to delivery.	X				
Lower development costs.	X				
More reusable code.	X				
Increased flexibility in development.	X				

**6. How would you rate the improvement (# of defects) in software quality due to Agile practices?**

Large improvement.

**7. How would you rate Cerner clients' satisfaction after the switch from Waterfall to Agile?**

Higher Satisfaction.

**8. What is the biggest benefit of Agile?**

shorter development cycles

**9. What is the biggest downfall of Agile?**

resourcing

**10. What is the biggest benefit of Waterfall?**

project scope control

### Survey Response 3

**1. Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?**

No Change

**2. Do you think the switch from Waterfall to Agile was beneficial to the team?**

No

**3. Do you think the switch from Waterfall to Agile was beneficial to the company?**

Yes

**4. Do you think the switch from Waterfall to Agile was beneficial to the clients?**

Yes

**5. The switch to Agile has...**

	Strongly Agree (1)	Agree (2)	No Difference (3)	Disagree (4)	Strongly Disagree (5)
Better met customer needs.		X			
Improved software quality.			X		
Faster time to delivery.		X			
Lower development costs.				X	
More reusable code.				X	
Increased flexibility in development.				X	

**6. How would you rate the improvement (# of defects) in software quality due to Agile practices?**

No improvement.

**7. How would you rate Cerner clients' satisfaction after the switch from Waterfall to Agile?**

Higher Satisfaction.

**8. What is the biggest benefit of Agile?**

happier clients

**9. What is the biggest downfall of Agile?**

Resourcing

**10. What is the biggest benefit of Waterfall?**

Resourcing

## Survey Response 4

**1. Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?**

Yes

**2. Do you think the switch from Waterfall to Agile was beneficial to the team?**

Yes

**3. Do you think the switch from Waterfall to Agile was beneficial to the company?**

Yes

**4. Do you think the switch from Waterfall to Agile was beneficial to the clients?**

Yes

**5. The switch to Agile has...**

	Strongly Agree (1)	Agree (2)	No Difference (3)	Disagree (4)	Strongly Disagree (5)
Better met customer needs.		X			
Improved software quality.		X			
Faster time to delivery.		X			
Lower development costs.		X			
More reusable code.		X			
Increased flexibility in development.		X			

**6. How would you rate the improvement (# of defects) in software quality due to Agile practices?**

Slight improvement.

**7. How would you rate Cerner clients' satisfaction after the switch from Waterfall to Agile?**

No Change.

**8. What is the biggest benefit of Agile?**

project scheduling

**9. What is the biggest downfall of Agile?**

Worse team morale

**10. What is the biggest benefit of Waterfall?**

nothing

## Survey Response 5

**1. Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?**

No Change

**2. Do you think the switch from Waterfall to Agile was beneficial to the team?**

No

**3. Do you think the switch from Waterfall to Agile was beneficial to the company?**

No

**4. Do you think the switch from Waterfall to Agile was beneficial to the clients?**

Yes

**5. The switch to Agile has...**

	Strongly Agree (1)	Agree (2)	No Difference (3)	Disagree (4)	Strongly Disagree (5)
Better met customer needs.		X			
Improved software quality.			X		
Faster time to delivery.			X		
Lower development costs.			X		
More reusable code.				X	
Increased flexibility in development.			X		

**6. How would you rate the improvement (# of defects) in software quality due to Agile practices?**

No improvement.

**7. How would you rate Cerner clients' satisfaction after the switch from Waterfall to Agile?**

No Change.

**8. What is the biggest benefit of Agile?**

clients perceived better code (met their functional requirements)

**9. What is the biggest downfall of Agile?**

resourcing...people work on domains they are not familiar with

**10. What is the biggest benefit of Waterfall?**

the right people are working on the right projects

## Survey Response 6

**1. Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?**

Yes

**2. Do you think the switch from Waterfall to Agile was beneficial to the team?**

Yes

**3. Do you think the switch from Waterfall to Agile was beneficial to the company?**

Yes

**4. Do you think the switch from Waterfall to Agile was beneficial to the clients?**

Yes

**5. The switch to Agile has...**

	Strongly Agree (1)	Agree (2)	No Difference (3)	Disagree (4)	Strongly Disagree (5)
Better met customer needs.		X			
Improved software quality.		X			
Faster time to delivery.		X			
Lower development costs.		X			
More reusable code.		X			
Increased flexibility in development.		X			

**6. How would you rate the improvement (# of defects) in software quality due to Agile practices?**

Slight improvement.

**7. How would you rate Cerner clients' satisfaction after the switch from Waterfall to Agile?**

Higher Satisfaction.

**8. What is the biggest benefit of Agile?**

**No Response**

**9. What is the biggest downfall of Agile?**

**No Response**

**10. What is the biggest benefit of Waterfall?**

**No Response**

## Survey Response 7

**1. Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?**

Yes

**2. Do you think the switch from Waterfall to Agile was beneficial to the team?**

Yes

**3. Do you think the switch from Waterfall to Agile was beneficial to the company?**

Yes

**4. Do you think the switch from Waterfall to Agile was beneficial to the clients?**

No

**5. The switch to Agile has...**

	Strongly Agree (1)	Agree (2)	No Difference (3)	Disagree (4)	Strongly Disagree (5)
Better met customer needs.				X	
Improved software quality.		X			
Faster time to delivery.			X		
Lower development costs.				X	
More reusable code.			X		
Increased flexibility in development.		X			

**6. How would you rate the improvement (# of defects) in software quality due to Agile practices?**

Slight improvement.

**7. How would you rate Cerner clients' satisfaction after the switch from Waterfall to Agile?**

Lower Satisfaction.

**8. What is the biggest benefit of Agile?**

higher quality software

**9. What is the biggest downfall of Agile?**

more turnover for projects

**10. What is the biggest benefit of Waterfall?**

**No Response**

## Survey Response 8

**1. Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?**

Yes

**2. Do you think the switch from Waterfall to Agile was beneficial to the team?**

Yes

**3. Do you think the switch from Waterfall to Agile was beneficial to the company?**

Yes

**4. Do you think the switch from Waterfall to Agile was beneficial to the clients?**

Yes

**5. The switch to Agile has...**

	Strongly Agree (1)	Agree (2)	No Difference (3)	Disagree (4)	Strongly Disagree (5)
Better met customer needs.		X			
Improved software quality.		X			
Faster time to delivery.		X			
Lower development costs.		X			
More reusable code.		X			
Increased flexibility in development.		X			

**6. How would you rate the improvement (# of defects) in software quality due to Agile practices?**

Slight improvement.

**7. How would you rate Cerner clients' satisfaction after the switch from Waterfall to Agile?**

Higher Satisfaction.

**8. What is the biggest benefit of Agile?**

**No Response**

**9. What is the biggest downfall of Agile?**

**No Response**

**10. What is the biggest benefit of Waterfall?**

**No Response**

## Survey Response 9

**1. Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?**

No Change

**2. Do you think the switch from Waterfall to Agile was beneficial to the team?**

Yes

**3. Do you think the switch from Waterfall to Agile was beneficial to the company?**

No

**4. Do you think the switch from Waterfall to Agile was beneficial to the clients?**

No

**5. The switch to Agile has...**

	Strongly Agree (1)	Agree (2)	No Difference (3)	Disagree (4)	Strongly Disagree (5)
Better met customer needs.		X			
Improved software quality.		X			
Faster time to delivery.			X		
Lower development costs.			X		
More reusable code.			X		
Increased flexibility in development.		X			

**6. How would you rate the improvement (# of defects) in software quality due to Agile practices?**

Slight improvement.

**7. How would you rate Cerner clients' satisfaction after the switch from Waterfall to Agile?**

Higher Satisfaction.

**8. What is the biggest benefit of Agile?**

**No Response**

**9. What is the biggest downfall of Agile?**

**No Response**

**10. What is the biggest benefit of Waterfall?**

**No Response**



## Survey Response 10

**1. Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?**

No

**2. Do you think the switch from Waterfall to Agile was beneficial to the team?**

No

**3. Do you think the switch from Waterfall to Agile was beneficial to the company?**

No

**4. Do you think the switch from Waterfall to Agile was beneficial to the clients?**

No

**5. The switch to Agile has...**

	Strongly Agree (1)	Agree (2)	No Difference (3)	Disagree (4)	Strongly Disagree (5)
Better met customer needs.				X	
Improved software quality.				X	
Faster time to delivery.				X	
Lower development costs.				X	
More reusable code.				X	
Increased flexibility in development.				X	

**6. How would you rate the improvement (# of defects) in software quality due to Agile practices?**

No improvement.

**7. How would you rate Cerner clients' satisfaction after the switch from Waterfall to Agile?**

No Change.

**8. What is the biggest benefit of Agile?**

nothing

**9. What is the biggest downfall of Agile?**

too much of a change for a corporation of this size.

**10. What is the biggest benefit of Waterfall?**

This worked well. There was no need to change.

## Survey Response 11

**1. Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?**

Yes

**2. Do you think the switch from Waterfall to Agile was beneficial to the team?**

Yes

**3. Do you think the switch from Waterfall to Agile was beneficial to the company?**

Yes

**4. Do you think the switch from Waterfall to Agile was beneficial to the clients?**

Yes

**5. The switch to Agile has...**

	Strongly Agree (1)	Agree (2)	No Difference (3)	Disagree (4)	Strongly Disagree (5)
Better met customer needs.	X				
Improved software quality.	X				
Faster time to delivery.	X				
Lower development costs.	X				
More reusable code.	X				
Increased flexibility in development.	X				

**6. How would you rate the improvement (# of defects) in software quality due to Agile practices?**

Large improvement.

**7. How would you rate Cerner clients' satisfaction after the switch from Waterfall to Agile?**

Higher Satisfaction.

**8. What is the biggest benefit of Agile?**

**No Response**

**9. What is the biggest downfall of Agile?**

**No Response**

**10. What is the biggest benefit of Waterfall?**

**No Response**

## Survey Response 12

**1. Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?**

Yes

**2. Do you think the switch from Waterfall to Agile was beneficial to the team?**

No

**3. Do you think the switch from Waterfall to Agile was beneficial to the company?**

Yes

**4. Do you think the switch from Waterfall to Agile was beneficial to the clients?**

Yes

**5. The switch to Agile has...**

	Strongly Agree (1)	Agree (2)	No Difference (3)	Disagree (4)	Strongly Disagree (5)
Better met customer needs.	X				
Improved software quality.		X			
Faster time to delivery.		X			
Lower development costs.				X	
More reusable code.		X			
Increased flexibility in development.				X	

**6. How would you rate the improvement (# of defects) in software quality due to Agile practices?**

Slight improvement.

**7. How would you rate Cerner clients' satisfaction after the switch from Waterfall to Agile?**

Higher Satisfaction.

**8. What is the biggest benefit of Agile?**

Smaller iterations / development cycles

**9. What is the biggest downfall of Agile?**

Team dynamics

**10. What is the biggest benefit of Waterfall?**

Maintain team relationship

## Survey Response 13

**1. Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?**

Yes

**2. Do you think the switch from Waterfall to Agile was beneficial to the team?**

Yes

**3. Do you think the switch from Waterfall to Agile was beneficial to the company?**

Yes

**4. Do you think the switch from Waterfall to Agile was beneficial to the clients?**

No

**5. The switch to Agile has...**

	Strongly Agree (1)	Agree (2)	No Difference (3)	Disagree (4)	Strongly Disagree (5)
Better met customer needs.			X		
Improved software quality.			X		
Faster time to delivery.			X		
Lower development costs.		X			
More reusable code.		X			
Increased flexibility in development.		X			

**6. How would you rate the improvement (# of defects) in software quality due to Agile practices?**

No improvement.

**7. How would you rate Cerner clients' satisfaction after the switch from Waterfall to Agile?**

No Change.

**8. What is the biggest benefit of Agile?**

More reusable code

**9. What is the biggest downfall of Agile?**

No real benefit for the customers

**10. What is the biggest benefit of Waterfall?**

**No Response**

## Survey Response 14

**1. Do you feel the Agile Software Methodology has had an overall positive effect on how software projects are progressing, compared to the Waterfall Software Methodology?**

Yes

**2. Do you think the switch from Waterfall to Agile was beneficial to the team?**

Yes

**3. Do you think the switch from Waterfall to Agile was beneficial to the company?**

Yes

**4. Do you think the switch from Waterfall to Agile was beneficial to the clients?**

Yes

**5. The switch to Agile has...**

	Strongly Agree (1)	Agree (2)	No Difference (3)	Disagree (4)	Strongly Disagree (5)
Better met customer needs.		X			
Improved software quality.		X			
Faster time to delivery.		X			
Lower development costs.		X			
More reusable code.					
Increased flexibility in development.		X			

**6. How would you rate the improvement (# of defects) in software quality due to Agile practices?**

Slight improvement.

**7. How would you rate Cerner clients' satisfaction after the switch from Waterfall to Agile?**

Higher Satisfaction.

**8. What is the biggest benefit of Agile?**

**No Response**

**9. What is the biggest downfall of Agile?**

**No Response**

**10. What is the biggest benefit of Waterfall?**

**No Response**