The Conceptual Model for Agile Tools and Techniques

Todd Haines
University of Maryland University College

Efosa C. Idemudia
Arkansas Tech University

Mahesh S. Raisinghani
Texas Woman's University

There have been limited studies that have developed Agile Techniques Model that adequately compared the pros and cons of various types of agile techniques. This research study develops a conceptual model that provides insights and understanding on agile techniques and methods to top management team members, software developers, project managers, and researchers. It develops the Agile Techniques Model to help investigate the pros and cons of various types of agile techniques. The theoretical background for our research model is the Stakeholder Theory. The results from our model support the conclusion that the agile project quality management method developed in the last fifteen years has proven to be useful, if not more useful, compared to the traditional method. However, the key objective of this research is to follow up on both the positive and negative factors to further test its probability. The information described in our model, proves that the agile project quality management method demonstrates both pros and cons on the techniques used within a project. The case studies from the literature in our research reiterate that agile methods increase productivity on the products for customers, but lacking communication is a common negative factor that stands out among all the cons.

INTRODUCTION

The emergence of the agile methods commenced in the mid-1990s (Highsmith 2002). Generally, agile methods improve product development performances (Sommer et al. 2015). Today we are living in a fast paced society with quick communication activities and demands. Traditional methods of project management are unidirectional and do not take into account a customer's wants and needs. However, the customer is ultimately the one to determine success or failure. Agile techniques on the other hand are dynamic getting regular input from the customer's. Based on research conducted, and according to Jamieson and Fallah (2012), the types of agile techniques will not only complete the requirements compared to other methodologies, but also improve the quality of the product(s), satisfy the customer(s), and display effective projects (p. 12). Parsons et al. (2007) argue that for the past three decades, the popularity of Agile software have increased drastically worldwide. Agile software has improved outcomes relating to competitive advantages, quality, satisfaction, and productivity for software development projects (Parsons et al. 2007). However, there have been limited studies that have developed

Agile Techniques Model that adequately compared the pros and cons of various types of agile techniques. To address these issues, our conceptual model provides insights and understanding on agile techniques to top management, software developers, project managers, and researchers. These agile techniques are not competing techniques, they just implement agility and ambidexterity in different scenarios. For instance, user acceptance testing is not competing with pair programming, instead both can be used to complement each other. In actual practice, multiple techniques are used simultaneously in projects

The theoretical background for our research model is the stakeholder theory. Also, as a project manager, one may want to consider a method that highly values the customer feedback. Throughout our paper, we will highlight the techniques of agile management and point out the pros and cons that one should consider. A brief overview of the pros and cons of agile techniques is illustrated in table 1.

TABLE 1
PROS AND CONS OF AGILE TECHNIQUES

Technique	Pros	Cons
Team involvement	• Customer's role is increased • More satisfaction and motivation • Further interactions	• Failure to agree on project goals • Communication
Continuous customer feedback	• Help initiate important information • Scrum which helps improve a product for a customer • Extreme Programming (XP) that formulates teams in regularly doing rework	• Wrong individual working with the project team presenting the wrong information • Not attempting for customer feedback • Obtaining comments too late
Pair programming	• Individuals and interactions act above different types of processes and tools • Improves product quality • Teams will show knowledge and courtesy	• Teaming up two individuals with the same expertise • Teaming an expert programmer with an average programmer • One partner relying heavily among there other partner
Automated acceptance testing	• Provides an understanding of any system needs • Reduces the rates of defects • Concurrence on the requirements	• Communication among customers • May not recognize failure • Difficulty to achieve automation
Refactoring code	• Design structure is changed • Developers can understand the system • Avoids errors	• No clue when it needs to be used • Time is undetermined • Can be costly

Technique	Pros	Cons
User acceptance testing	• Customer(s) determine if the software is ready • Guaranteed higher quality • Software requirements are up-to-date • Cost-effective procedure	• User stories are not developed • Wrong input into a system and will be more costly • Customer doesn't lead the testing
Meetings	• Members share what they have accomplished • Projects can finish sooner rather than later • Develops quality for both the current project and any future projects	• No commonality among each member on the team • Meetings are extended longer than planned • Not communicating • Poor management

Note. Information for the pros and cons on the techniques are from Alshehri & Benedicenti (2014), Biju (2008), Bjarnason, Wnuk, & Regnell (2012), Denning (2012), Dimitrijevic, Jovanovic, & Devedzic (2015), El-Attar & Miller (2010), Foster (2013), Ghourbanpour & Yektaie (2011), Grapenthin, Poggel, Book, & Gruhn (2015), Gupta, Bhattacharya, & Singha (2013), Hasnain & Hall (2008), Haugset & Stalhane (2012), Hoda, Noble, & Marshall (2011), Jamieson & Fallah (2012), Nanau (2010), Olsson, Alahyari, & Bosch (2012), Plonka, Sharp, Van der Linden, & Dittrich (2015), Randall (2014), Tessem (2014), Unterholzner (2014), Van Waardenburg & Van Vliet (2013), and Yu & Petter (2014).

LITERATURE REVIEW AND SYNTHESIS

The importance of agile project quality management has grown significantly in the project community, as forecasted by the software developers who discussed the beginning of the agile methodology back in February 2001. According to Tessem (2014), the agility proponents combined have constituted the development processes that are different compared to the traditional method, and appear to provide higher quality in software at a cheaper cost (p. 873). Individuals that lead an agile project team help set the direction, align individuals, obtain all resources, and provide motivation. In addition, "agile teams are self-organizing teams" (Hoda et al., 2011, p. 522). These teams consist of individuals that control their own work, adjust work based on needs, and communicate with team members on decisions.

The main focus in the agile method is customer satisfaction (Parsons et al 2007). According to Biju (2008), the agile method is based on the following standards:

- "individuals and interactions over processes and tools;
- working software over comprehensive documentation;
- customer collaboration over contract negotiation;
- responding to change over following a plan" (p. 97).

Even though there are positive outcomes of the agile method, there are also disadvantages (Gregory et al. 2014; Tomanek et al. 2014). Using this method for the first time is challenging to the project team because it requires members to change their existing process/methodology and learn new rules of this agile methodolody (Biju, 2008, p. 101). In addition, more rework may be involved among the project team members, but it can be avoided if requirements analysis is performed properly (Biju, 2008, p. 101).

The framework proposed in this paper aims to address the pros and cons of the techniques mentioned below that are used within the agile methodology. To supplement the research of agile methodology developed by the software developers in February 2001, this paper proves that the agile methodology can be beneficial and challenging in certain circumstances for software project management.

THE STAKEHOLDER THEORY

Our research model is shown in *Figure 1* and the theoretical background for our research model is the stakeholder theory. Stakeholder theory posits that a firm's performance depends on how management and key decision makers address stakeholders' interests (Donaldson and Preston 1995; Freeman 2010; Frooman 1999; Mitroff and Linstone 1993). In their work, Mitroff and Linstone (1993), argue that stakeholder theory is a critical theory that should be used to study all managerial and organizational issues because all decisions involve assumption analysis. Hence, the stakeholder theory can be used to study important managerial and organizational issues such as selecting the best strategy to improve technology effectiveness, making effective use of data resources, developing IS human resources, improving systems quality, improving service quality, and evaluating software and hardware effectiveness all through the lens of the stakeholders' interests (i.e., employees, customers, communities, governments, and suppliers). Stakeholder theory gives management and key decision-makers the opportunity to challenge, question, and critique their assumptions before selecting the best strategy (Donaldson and Preston 1995; Mitroff and Linstone 1993). Stakeholder theory provides management and CEOs with insights on how to respond to external crises such as global recession and the resultant credit crunch. Phillip (2004) argues that stakeholder theory helps management and key decision-makers in MIS firms to select the best strategy by addressing the following managerial issues:

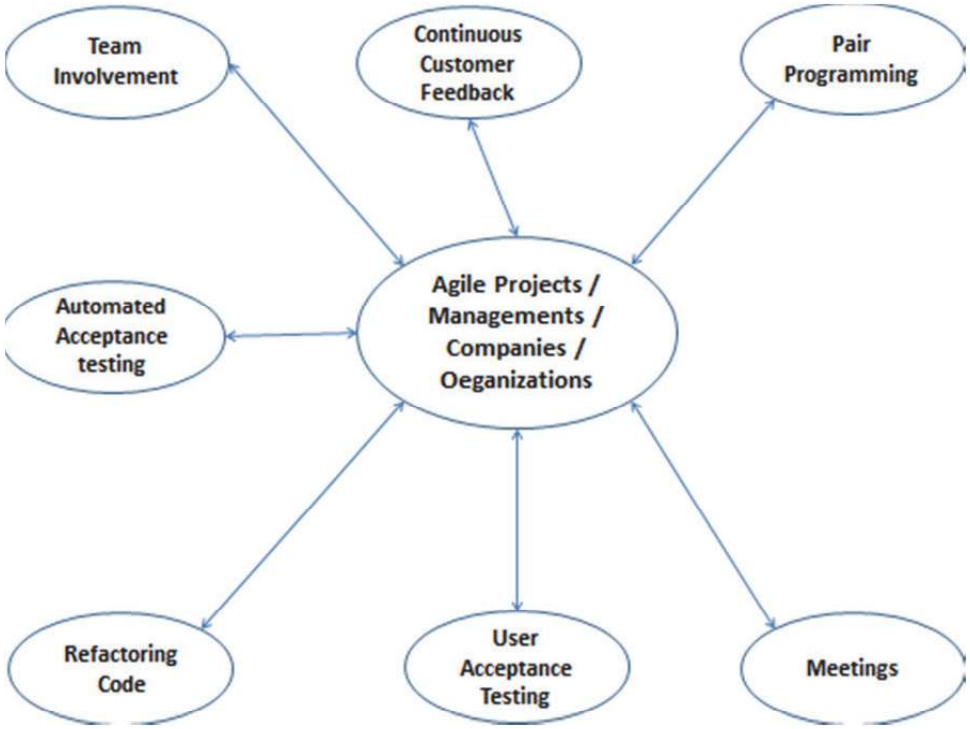
1) Why should managers and key decision makers pay interest and attention to the stakeholders of information technology (IT) firms?
2) What do IT/management information systems (MIS) stakeholders' desire and need?
3) How should the desires, requests, and needs of all stakeholders in the IT/MIS firms be ranked and prioritized?
4) Are the morals, principles, values, ethics, and beliefs of the IT/MIS firms good or bad?

Overall, the stakeholder theory gives management and key decision makers the opportunity to develop strategies to adapt to stakeholders' interests and influences.

THE AGILE TECHNIQUES MODEL

Our research model posits that agile techniques are most effective and efficient if all the techniques are considered before selecting any specific technique and project. The arrows in *Figure 1* are multidirectional, meaning that if any of the constructs change (i.e., agile techniques), the project management, processes, people, and procedures must adapt to those changes. Failing to adapt to the changes leads to an execution gap. Keziere (2006) defines an execution gap as the gap between a company's execution and management goals. Keziere (2006) argues that the execution gap hinders managements' performances because of their companies' inability to respond to the market, which decreases both the company's market share and profits. In conclusion, changes in any of the stakeholder's interests affect the whole structure as shown in *Figure 1*. Below we are going to present all the components and techniques that make up our research model.

FIGURE 1
THE AGILE TECHNIQUES MODEL



Team Involvement

Team involvement includes regular participation among each team member in five ways. Participation includes the following: members decide how their work is done, make suggestions for improvement, set goals, planning, and monitor their performance ("What is employee involvement?," 2015). Team involvement is a beneficial concept because if all individuals are actively involved in a process then it's likely that the team will master it (Andries and Czarnitzki 2014).

Pros. According to Hoda et al. (2011) customers involved in a software development project are vital and is one of the major factors in agile software development (p. 521). This method helps spread out the customers' roles within the whole development process (Hoda et al., 2011, p. 521). The interaction between the customer and the developer demonstrates good team involvement. Another pro in team involvement is empowerment (Tessem 2014). Empowerment is considered to be "central to agile software development organization's overall approach" (Tessem, 2014, p. 874) and with each individual on the team being empowered, he/she will achieve more job satisfaction and be more motivated (Tessem, 2014, p. 873). Another benefit of having team involvement is including further interactions with the customer(s), which will improve the requirements definition and project scope definition/management (Jamieson & Fallah, 2012, p. 15).

Cons. Even though there have been a large number of projects adopting and successfully applying agile techniques, it seems complicated for teams to connect them to traditional techniques (Jamieson & Fallah, 2012, p. 12). One negative aspect of team involvement is by the stakeholders failing to agree on the overall project goals, which can cause an excessive burden on the scope of the project (Bjarnason, Wnuk, & Regnell, 2012, p. 1108). In addition, communication among team members and other colleagues is another negative aspect involved in agile project quality management (Bjarnason et al. 2012). According

to Van Waardenburg & Van Vliet (2013, p. 2159), communication is very difficult, especially if individuals are geographically separated, not in the same country, and/or speak different languages.

Continuous Customer Feedback

Continuous customer feedback includes information that is provided by the customer and they provide either their liking or frustration with a finished project. The comments and complaints provided by the customer(s) to a project team are vital to improve and concentrate on the desires and requests of the customer(s). Customer feedback can be obtained through written or oral surveys, online forms, emails, letters, phone calls, or face-to-face meetings to the project team ("What is customer feedback?," 2015).

Pros. According to Jamieson and Fallah (2012), feedback provided by customers on a continuous basis is beneficial by allowing relay of important information to the team on new requirements, verifying that the requirements are executed properly, and any probable issues that arise (p. 15). This technique easily helps project teams adapt and improve the applications to meet the needs of all customers (Jamieson & Fallah, 2012, p. 15). In agile project quality management, a management framework method called Scrum is a set of rules that project teams follow and based on customer feedback; improve a product for a customer (Randall, 2014, p. 27). Another method to obtain continuous customer feedback is called Extreme Programming (XP). This method formulates teams in regularly doing rework to the project by obtaining feedback from the customer(s) (Biju, 2008, p. 98).

Cons. One of the negative aspects of customer feedback is potentially having the wrong individual interacting with the project team and presenting inaccurate information (Jamieson & Fallah, 2012, p. 15). Projects based on inaccurate information will ultimately lead to unsuccessful development. In addition to this negative phase, another unhelpful outcome of continuous customer feedback is failing to see what the customer(s) exactly wants (Denning, 2012, p. 24). Not allowing the customer to speak during the project will cause the team to develop unusable products. Lastly, another con with continuous customer feedback is having a team obtain feedback from customers too late in the project (Olsson et al., 2012, p.393). If the team does not receive the feedback from the customer(s) in a timely fashion, the project will not be successful.

Pair Programming

Pair programming, which can also be called peer programming, involves two programmers that work together in one area ("Pair programming," 2015). One individual writes the code, while the other individual overlooks every line of the code. Each individual will commonly switch their roles throughout the project. The individual that is writing the code focuses all of their concentration on the characteristics of completing the current task while using the other individual as a backup. The individual that is overlooking the code can also judge the strategic direction of the work being performed, by coming up with the ideas for improvement and possible future problems to address ("Pair programming," 2015).

Pros. According to Gupta et al. (2013), pair programming within agile project quality management involves having individuals and interactions act above different types of processes and tools (p. 108). This principle creates a more professional environment allowing two people to work better together. In addition, another positive aspect of pair programming is that this technique can improve product quality by having fewer defects in comparison to the output by an individual working alone (Jamieson & Fallah, 2012, p. 15). One dimension of quality that demonstrates improvement on a product is quality assurance. According to Foster (2013), quality assurance is when individuals demonstrate knowledge and courtesy to the customer, and build trust and confidence with that customer (p. 6).

Cons. One negative aspect among pair programming within agile project quality management is teaming up two individuals with the same expertise as this will cause counter-productive work among one another (Plonka, Sharp, Van der Linden, & Dittrich, 2015, p. 67). In addition, another pessimistic fact of pair

programming is by placing an expert programmer with an average programmer. An expert programmer could help increase the skill level of an average programmer, but if that individual is not interested in increasing this/her knowledge, conflict/s could arise (Alshehri & Benedicenti, 2014, p. 2469). Lastly, one may find a partner relying heavily on the other, which could potentially keep one from learning new methods (Gupta et al., 2013, p. 112).

Automated Acceptance Testing

By definition, this technique is a formal description on the behavior of a software product, which can be articulated as a model or a procedure ("Guide to Agile," 2013). A different number of details and approaches can be used for such models or procedures, but in many situations, the goal is to execute tests using a software tool, either ad-hoc to the project development team or an off-the-shelf product ("Guide to Agile," 2013). An acceptance test is generally understood to have a pass or fail result.

Pros. According to Haugset and Stalhane (2012), one development method within agile project quality management that has been expanding is automated testing (p. 5289). With this method used, it helps provide an understanding of system needs and automatically tests software at a company level. Another positive factor of automated testing will be that it helps reduce the rate of tracing and fixing defects (Jamieson & Fallah, 2012, p. 16). This aspect will also help keep the project moving at the right pace. In addition, this type of testing allows both the customer and the developer to agree on the requirements written during the testing phase (Haugset & Stalhane, 2012, p. 5294). By agreeing on the requirements, this method would decrease the uncertainty and enhance the stability of the tests (Haugset & Stalhane, 2012, p. 5294).

Cons. One negative aspect of automated testing is communication among customers (Haugset & Stalhane, 2012, p. 5294). All customers focus on different topics and if they do not discuss this with the developers, then the system will not work properly. Another shocking negativity is that this tool may not recognize when tests fail (Nanau, 2010, p. 185). In the event that this happens, a project could keep going forward until the customer(s) passes it to the attention of the project team. At this point, the project may fall behind schedule if the problem is discovered later in the process. Finally, another negative aspect of automated testing on agile project quality management could be its difficulty to achieve automation (Nanau, 2010, p. 187). If this is to occur, the testing effort and time will increase.

Refactoring Code

This technique is a process to reform the present computer code without changing the exterior performance. Refactoring will help improve the nonfunctional characteristics of the software being used ("Code refactoring," 2015). The advantages of this technique "include improved code readability and reduced complexity, which can improve source code maintainability and create a more expressive internal architecture or object model to improve extensibility" ("Code refactoring," 2015).

Pros. According to Jamieson and Fallah (2012), refactoring code on an agile project is where a design structure is changed, but the aspect of the code is not changed which can be accomplished more successfully as compared to a traditional project (p. 17). By changing the structure, developers have a better understanding of the system, are able to interpret it and maintain it better, and utilize it more efficiently (Ghourbanpour & Yektaie, 2011, p. 6). In addition to those positive factors, another constructive feature of refactoring code is it helps reduce a developer's effort and also avoids errors (Unterholzner, 2014, p. 70). This effort is accomplished by manually changing the code of the design structure.

Cons. One drawback of this technique is that programmers may not know when to apply it (Ghourbanpour & Yektaie, 2011, p. 6). The uncertainty of its use may restrict programmers to utilize the refactoring code. Another negative feature is the uncertainty of the required time involved for a

programmer to change the structure. Time involved would be based on the programmer's experience and the type of programming language that is being used (Ghourbanpour & Yektaie, 2011, p. 6). Lastly, this type of technique can be costly if it is done incorrectly.

User Acceptance Testing

In this technique, the users of the software test the software to ensure it can withstand the needed tasks in real-world scenarios, and within the specifications ("What is user acceptance testing?", 2015). User acceptance testing also is one of the last and most significant software project techniques that must take place before new developed software is put to the market ("What is user acceptance testing?", 2015). Lastly, "user acceptance testing is also known as beta testing, application testing or end user testing" ("What is user acceptance testing?", 2015).

Pros. Within agile project quality management, this technique involves having the customer(s) determine if the software is ready to be released to the company (Jamieson & Fallah, 2012, p. 17). This is one positive method that all companies should use when developing an agile project. In addition, this technique should be used in several sessions throughout the project to guarantee higher product quality. Stakeholders also provide regular feedback during testing which ensures that software requirements are up-to-date and useful throughout the whole process (Dimitrijevic, Jovanovic, & Devedzic, 2015, p. 353). This technique is a cost-effective procedure because it allows the customer(s) to articulate the requirements for a system through testing (El-Attar & Miller, 2010, p. 285). This will help the developers build a better system that can meet the customer's expectations and requirements.

Cons. One negative aspect of user acceptance testing is that if the users do nott include their user stories and the testing only includes stories that the development team has created, then this tool will be unsuccessful. Another pessimistic feature of this technique is if the team injects the wrong input into a system, different output will be produced (El-Attar & Miller, 2010, p. 292). Obviously, the user(s) will not accept what has been produced, but it will be more costly to develop the proper system. Lastly, this technique will not be successful if the customer does not lead this testing (Jamieson & Fallah, 2012, p. 17).

Meetings

Meetings are considered to be formal or informal planned assemblies of individuals to ponder on particular issues and problems, and to make final choices ("What is meeting?," 2015). Formal meetings have a specific agenda and are more structured with a definite time, place, and duration. Meetings can be divided into two groups, an organizational meeting or an operational meeting. An organizational meeting involves shareholders and management whereas an operational meeting involves management and team members ("What is meeting?," 2015).

Pros. When a team is using agile project quality management, one thing they need to do in order to be successful is conduct regular meetings. This continuity will allow the team members to share their project contributions and resolve any issues that may interfere with them not achieving their objectives (Jamieson & Fallah, 2012, p. 18). According to Grapethin et al. (2015), there is one agile model that helps teams construct projects into iterations, and that is called Scrum (p. 1). Scrum consists of the following four steps: iteration planning, implementing and testing a certain product increment, results being presented to the stakeholders, and each member on the team revealing the work they have accomplished to improve future sprints (Grapethin et al., 2015, p. 1). The last step seems to be the most crucial one of all. All members of the team need to reveal the work they have accomplished in order to complete the project on schedule. By each member of the team doing this step, they will be able to develop quality for both the current project and any future projects (Jamieson & Fallah, 2012, p. 18).

Cons. One of the negative aspects of holding team meetings is by having no commonality among each member on the team (Yu & Petter, 2014, p. 912). This would include members not knowing what tasks they have been assigned and relationships among each member. Another unconstructive method of holding meetings is if the meetings are extended longer than planned. Individuals not communicating in the meeting and or wrongful information presented during the meeting could also cause a challenge (Yu & Petter, 2014, p. 915). In addition to those negative elements of holding meetings, another one would be poor management. Poor management would involve holding the meetings where the developers would not be able to get their point across, but according to Hasnain and Hall (2008), "developers now have them so [they] know what's going on" (p. 1), but management is not involved. This would adversely affect the project management because they would not know what was being discussed.

IMPLICATIONS FOR RESEARCH

Contributions Of This Study

The information compiled in our study gives a general overview of the Agile Techniques Model and the pros and cons of the agile techniques used within agile project quality management. Our conceptual model provides insight, understanding, and reference for program managers, top managements, project managers, and researchers to utilize the techniques mentioned. In today's workforce, team involvement is a huge factor in regular business interactions. Generally, in most careers, the assignments, meetings, and interactions performed have always been group focused. In addition, team involvement has helped individuals achieve more satisfaction with the job and obtain more motivation. Another major element in today's workforce is continuous customer feedback. In our current jobs, we constantly communicating with customers to improve and address the needs and wants they currently have on products and services. However, there have been times when communication hasn't occurred between some organizations/companies and their customers, and this has led to unsuccessful projects. Lastly, we feel in today's environment, meetings being held are crucial for any company or organization to succeed. In our current work roles, our offices are constantly conducting meetings. The meetings help all individuals share their contributions and solve any issues that arise. If our offices didn't conduct weekly meetings, we feel projects wouldn't be successful.

Limitations Of This Study

Based on our research, there were several limitations. Our study on the pros and cons on the types of agile techniques was limited on a small trial size. The trial size could have been increased by collecting the data earlier which would have enlarged the amount of information presented and benefited the results. Also, we did not test the bidirectional arrows in our research model, the Agile Techniques Model, and we encourage future researchers to empirically validate our research model.

Directions For Future Research

Based on the agile method being used for only the last 15 years, researchers interested in this topic should do further research. Areas that require in depth research should include the following: conducting surveys, performing lab tests, researching other individuals' papers, and communicating with large and small companies that have attempted this method to see if it has succeeded or failed. Researchers should focus heavily on these techniques, but also see if any others can be used within this process. In addition, individuals should see if one outcome outweighs the other. Finally, researchers should empirically validate our model and rank the constructs in our model based on their importance and significant.

IMPLICATIONS FOR MANAGEMENT

Key Takeaways

Managers need to be aware that the agile method in our model is rarely new, and that there are both pros and cons upon this method being utilized. However, managers also need to realize that this method requires the following: self-organizing teams to complete work based on the customers' needs, teams that interact regularly with the customers, work being performed is preceded in a repetitive technique, and is progressed towards meeting the fulfillment of the customers' needs (Denning, 2015, p. 11). If one or more of these suggestions is not used, then this method will be difficult to accomplish a successful project. Also, top managers, project managers, and software developers should know that the agile projects should adapt to these techniques (i.e., if the agile project changes then the technique should adopt to those changes).

Lessons Learned

Based on our research, we learned a decent amount of information on both the pros and cons of the techniques used within the agile method. The number one positive factor on the techniques mentioned was increased productivity on the products for customers. A few pros that help increase the productivity is having the customers more involved, communicating further with customers, and having better involvement among the team members. As for the negative factor, there seems to be one that stands out among all the cons stated throughout the paper, and that is lack of communication. The first stage of communication begins with the team and if the members don't speak to one another, then the project will fail in the very beginning. In addition, if the customers are not communicated with on a regular basis, then there will be a finished product, but that product will most likely be refused by the customers. Based on the research conducted, agile project quality management has both the positive and negative factors. This methodology should be researched more among corporations to view if it is a better tool to use compared to the traditional methodology used by the company.

CONCLUSION

Due to the agile method being used within the last 15 years, and the information provided throughout our paper, Individuals should conduct research about this method by applying the following techniques mentioned above: team involvement, continuous customer feedback, pair programming, automated acceptance testing, refactoring code, user acceptance testing, and meetings. From our research, we highlighted both the pros and cons of certain techniques used within the agile method. The pros of agile project quality management identified during the research demonstrates that customer's roles are increased from the beginning to the end of a project, product quality is improved, the defect rate is reduced, errors are avoided, cost-effective procedures are performed, and projects can be completed on schedule. Some of the negative factors may include lack of communication/customer feedback, partnering with the wrong individuals for requirements definition, increased project costs, and poor project management. With the information gathered, every project team should look into both the agile project quality management method and the traditional methods. By doing so, the team can view which method will be most beneficial in obtaining quality improvement on the product(s), customer satisfaction, and effective projects.

REFERENCES

Alshehri, S., & Benedicenti, L. (2014). Ranking and rules for selecting two persons in pair programming. Journal of Software, 9(9), 2467-2473.

Andries, P., & Czarnitzki, D. (2014). Small firm innovation performance and employee involvement. Small Business Economics, 43(1), 21-38.

Biju, S.M. (2008). Agile software development. *E-Learning, 5*(1), 97-102.

Bjarnason, E., Wnuk, K., & Regnell, B. (2012). Are you biting off more than you can chew? A case study on causes and effects of over scoping in large-scale software engineering. Information and Software Technology, 54, 1107-1124. doi:10.1016/j.infsof.2012.04.006

Code refactoring. (2015). Retrieved September 29, 2015 from the Wikipedia: https://en.wikipedia.org/wiki/Code_refactoring

Denning, S. (2012). How Agile can transform manufacturing: The case of Wikispeed. Strategy & Leadership, 40(6), 22-28. doi:10.1108/10878571211278859

Denning, S. (2015). Agile: It's time to put it to use to manage business complexity. Strategy & Leadership, 43(5), 10-17. doi:10.1108/SL-07-2015-0057

Dimitrijevic, S., Jovanovic, J., & Devedzic, V. (2015). A comparative study of software tools for user story management. Information and Software Technology, 57, 352-368. doi:10.1016/j.infsof.2014.05.012

Donaldson, T. & Preston, L. 1995. The stakeholder theory of the modern corporation: Concepts, evidence and implications. Academy of Management Review 20, 65-91

El-Attar, M., & Miller, J. (2010). Developing comprehensive acceptance tests from use cases and robustness diagrams. Requirements Engineering, 15(3), 285-306. doi:10.1007/s00766-009-0088-6

Foster, S.T. (2013). Managing quality: Integrating the supply chain (5th ed.). Upper Saddle River, NJ: Pearson Education, Inc.

Freeman, R. E. (2010). Strategic management: A stakeholder approach. Cambridge University Press.

Frooman, J. (1999). Stakeholder influence strategies. Academy of management review, 24(2), 191-205.

Ghourbanpour, D.K., & Yektaie, M.H. (2011). Towards pattern-based refactoring: Abstract factory. International Journal of Advanced Research in Computer Science, 2(3), 6-9.

Grapenthin, S., Poggel, S., Book, M., & Gruhn, V. (2015). Improving task breakdown comprehensiveness in agile projects with an interaction room. Information and Software Technology, 67, 254-264. doi:10.1016/j.infsof.2015.07.008

Gregory, P., Plonka, L., Sharp, H., & Taylor, K. (2014). Bridging the gap between research and practice: The agile research network. Paper presented at the 161-168.

Guide to Agile practices. (2013). Retrieved September 29, 2015 from http://guide.agilealliance.org/guide/acceptance.html

Gupta, S., Bhattacharya, V., & Singha, M. (2013). Pair programming "potential benefits and threats". International Journal of Advanced Computer Research, 3(1), 108-113.

Hasnain, E., & Hall, T. (2008). Preliminary investigation of stand-up meetings in agile methods. IAENG International Journal of Computer Science, 35(4), 1-3.

Haugset, B., & Stalhane, T. (2012). Automated acceptance testing as an agile requirements engineering practice. 2012 45th Hawaii International Conference on System Sciences, 5289-5298. doi:10.1109/HICSS.2012.127

Highsmith, J. A. (2002). Agile software development ecosystems (Vol. 13). Addison-Wesley Professional.

Hoda, R., Noble, J., & Marshall S. (2011). The impact of inadequate customer collaboration on self-organizing agile teams. Information and Software Technology, 53, 521-534. doi:10.1016/j.infsof.2010.10.009

Jamieson, J.M., & Fallah, M.H. (2012). Agile quality management techniques. Software Quality Professional, 14(2), 12-21.

Mitroff, I. I., & Linstone, H. A. (1993). The unbounded mind: Breaking the chains of traditional business thinking. Oxford University Press.
Keziere, R. (2006). Closing the Business Process Management Execution Gap. BUSINESS INTELLIGENCE JOURNAL, 11(2), 39.
Nanau, C. (2010). Automated test generation and agile methodologies. Bulletin of the Transylvania University of Brasov, Series III: Mathematics, Informatics, Physics, 3(52), 183-190.
Olsson, H.H., Alahyari, H., & Bosch J. (2012). Climbing the "stairway to heaven" – A multiple-case study exploring barriers in the transition from agile development towards continuous deployment of software. 2012 38th Euromicro Conference on Software Engineering & Advanced Applications, 392-399. doi:10.1109/SEAA.2012.54
Pair programming. (2015). Retrieved September 29, 2015 from the Wikipedia: https://en.wikipedia.org/wiki/Pair_programming
Parsons, D., Ryu, H., & Lal, R. (2007). The impact of methods and techniques on outcomes from agile software development projects. In Organizational Dynamics of Technology-Based Innovation: Diversifying the Research Agenda (pp. 235-249). Springer US.
Plonka, L., Sharp, H., Van der Linden, J., & Dittrich, Y. (2015). Knowledge transfer in pair programming: An in-depth analysis. International Journal of Human-Computer Studies, 73, 66-78.
Phillips, R. A. (2004). Some key questions about stakeholder theory. Ivey Business Journal.
Randall, R. M. (2014). Agile at IBM: Software developers teach a new dance step to management. Strategy & Leadership, 42(2), 26-29. doi:10.1108/SL-01-2014-0003
Sommer, A. F., Hedegaard, C., Dukovska-Popovska, I., & Steger-Jensen, K. (2015). Improved product development performance through Agile/Stage-gate hybrids: The next-generation stage-gate process? Research Technology Management, 58(1), 34-44.
Tessem, B. (2014). Individual empowerment of agile and non-agile software developers in small teams. Information and Software Technology, 56, 873-889. doi:10.1016/j.infsof.2014.02.005
Tomanek, M., Cermak, R., & Smutny, Z. (2014). A conceptual framework for web development projects based on project management and agile development principles. Paper presented at the 550-558.
Unterholzner, M. (2014). Improving refactoring tools in Smalltalk using static type inference. Science of Computer Programming, 96, 70-83. doi:10.1016/j.scico.2013.11.032
Van Waardenburg, G., & Van Vliet, H. (2013). When agile meets the enterprise. Information and Software Technology, 55, 2154-2171. doi:10.1016/j.infsof.2013.07.012
What is customer feedback? (2015). In BusinessDictionary online. Retrieved September 29, 2015 from http://www.businessdictionary.com/definition/customer-feedback.html
What is employee involvement? (2015). In BusinessDictionary online. Retrieved September 29, 2015 from http://www.businessdictionary.com/definition/employee-involvement.html
What is meeting? (2015). In BusinessDictionary online. Retrieved September 29, 2015 from http://www.businessdictionary.com/definition/meeting.html
What is user acceptance testing (UAT)? (2015). In Techopedia online. Retrieved September 29, 2015 from https://www.techopedia.com/definition/3887/user-acceptance-testing-uat
Yu, X., & Petter, S. (2014). Understanding agile software development practices using shared mental models theory. Information and Software Technology, 56, 911-921. doi:10.1016/j.infsof.2014.02.010