CUSTOMER INTERACTION IN SOFTWARE DEVELOPMENT: A COMPARISON OF SOFTWARE METHODOLOGIES DEPLOYED IN NAMIBIAN SOFTWARE FIRMS

Gloria E. Iyawa
University of Namibia
Namibia
gloria.iyawa@gmail.com

Marlien E Herselman
Meraka Institute, CSIR &
University of South Africa
South Africa
MHerselman@csir.co.za

Alfred Coleman
University of South Africa
South Africa
Colema@unisa.ac.za

ABSTRACT

Software methodologies provide guidelines for the development of software applications. Studies reveal that customer interaction in the software development process improves the chances that software applications will meet customers' needs. Despite a number of software methodologies introduced and a comparison of these methodologies, there is a dearth of studies that empirically investigate customer interaction between these software methodologies within the Namibian context. The purpose of this study was to examine the differences in customer interaction between software methodologies deployed in Namibian software firms. The study adopted a qualitative, case study approach. Data was collected through standardized, open-ended interviews. The findings show that the methodologies deployed in Namibian software firms include the waterfall model, Scrum, iterative model, eXtreme Programming (XP), and rapid application development (RAD). The findings also reveal that although there was in-depth customer interaction in Scrum, the iterative model, XP and RAD, customer interaction in the software development process could also be challenging. The findings provide useful insights in software methodologies deployed in Namibian software firms and the experience within the Namibian context. An implication for software project managers and software developers is that customer interaction should be properly managed to ensure that the software methodologies for improving software development processes are effectively deployed.

Keywords: Waterfall model, Scrum, iterative model, eXtreme Programming, rapid application development, Namibia.

1. INTRODUCTION

Software organisations deploy different software methodologies in the software development process, selected by software teams based on a variety of reasons (Young, 2013). While different software development methodologies have emerged, each with its own strengths and weaknesses (Bassil, 2012), agile methodologies emphasise customer interaction in the software development process (Williams & Cockburn, 2003). In agile methodologies, the emphasis is on developers that closely interact with customers in the software development process (Williams & Cockburn, 2003). Agile software methodologies have been widely adopted by developers after its introduction in 2001, with eXtreme Programming (XP) and Scrum being the most popular agile software methodologies (Dingsøyr et al., 2012). The involvement of customers in the development process ensures that software applications perform according to customers' expectations.

Software methodologies are an important aspect in software development companies. Maddison (1984) defines a methodology as a "recommended collection of philosophies, phases, procedures, rules, techniques, tools, documentation, management, and training for developers of information systems". Hence, understanding the differences in customer interaction between software methodologies is not only important to the software team but also important for improving software processes. Researchers have elaborated on customer interaction in the requirements gathering phase of the software development process (Keil &

Carmel, 1995; Saiedian & Dale, 2000). Customer interaction in software development has been well documented (Keil & Carmel, 1995; Fogelstrom et al., 2010). Studies elaborating on the reasons for inconsistencies and reduced customer interaction in a software development project have also been well documented (Lohan et al., 2011; Akinnuwesi et al., 2013; Safwan et al., 2013). Furthermore, various studies have emphasised the effects of customer interaction in software projects where agile software methodologies have been adopted (Grisham & Perry; 2005; Dybå & Dingsøyr, 2008; Wang et al., 2008; Lohan et al., 2011). Different studies have compared the different software methodologies available. For example, Awad (2005), Moniruzzaman & Hossain (2013), and Javanmard & Allan (2015) have compared agile and traditional software methodologies. A study by Munassar & Govardhan (2010) compares the traditional waterfall model, iterative model, v-shaped model, spiral model and XP. A recent study by Chandra (2015) also compared different software development methodologies.

Despite these studies, there is a dearth of studies empirically comparing customer interaction between the different software methodologies within the contexts of developing countries like Namibia. Furthermore, academic literature on software development methodologies deployed in Namibian software firms is almost non-existent. This study hence contributes to the meagre literature on software development methodologies focusing on the Namibian context and seeks to provide empirical findings on the differences in customer interaction between the software methodologies deployed in Namibian software firms. The findings of this study are expected to inform software project managers and software developers on the employment and experience of software methodologies within the Namibian context.

1.1 Research Objectives

The objectives of this study were:

- To investigate the different software methodologies deployed in Namibian software firms.
- To identify the differences in customer interaction between the software methodologies deployed in Namibian software firms.

The rest of this article is structured as follows: related work, research methodology, results, discussion and conclusions.

2. Related Work

Every field of "business" requires customers to make use of goods and services to survive. There are many definitions for the term "customer", including Riley's (2012) definition, which describes a customer as an individual who benefits from a commodity for which he/she may have made financial sacrifices. A more detailed definition from the Business Dictionary (2014) explains that a customer is an individual who decides to use a service after a choice has been made from various service providers. This definition suggests that an individual benefits from a commodity after making financial sacrifices, which means the customer has the option to choose from among several service providers. Hence, service providers have to be competitive in their service delivery to persuade customers to purchase their product. In the context of software development, customers should be sufficiently satisfied with a software product to patronise the same software firm for another project, and this can arguably be achieved when developers and customers effectively work together.

Customer interaction has not only been discussed in software development contexts (Humphreys & Grayson, 2008; Wattanakamolchai, 2008; Sun et al., 2010; Hoyer et al., 2010; Terblanche, 2014). These studies reveal that customer involvement in the production of goods and services is fundamental to making significant improvements in the product output.

Customer involvement has been identified as one of the key factors for successful software projects (Tanner & Willingh, 2014). Wang et al. (2008) indicate that customers become involved in the process by giving constant feedback to developers and testing each software release. The current literature also reveals that customer interaction in software development has been studied in developing contexts. A study by Akinnuwesi et al. (2013) analyses the factors that inhibit customer involvement in software development projects within the Nigerian context. Akinnuwesi et al. (2013) explored the level to which end users participate in software projects. The findings reveal that a lack of computer literacy, age, gender and cultural issues are among the factors that play an active role in reducing customer involvement in the software development process.

Furthermore, Safwan et al. (2013) investigated how the agile software methodology is perceived in Sri Lanka, a developing country. The findings reveal that software development teams perceive agile methodologies to be useful. However, these software teams experience challenges when adopting agile methodologies, including the “difficulty of getting everyone in the team to take responsibilities” (Safwan et al., 2013:6). In addition, Haikin (2013) suggests that there is a huge potential in applying agile methodologies within information and communication technologies for development (ICT4D) projects.

Anderson & Mϕrch (2013) and Franken et al. (2015) suggest customer involvement using social media as a mechanism for facilitating customer interaction in the software development process. However, customer interaction can be achieved through the software methodologies deployed. Different studies suggest agile methodologies as methodologies that facilitate customer interaction (Lohan et al., 2011; Sharma et al., 2012). Agile methodologies include “XP, Scrum, Crystal Clear, Feature Driven Development (FDD), Lean Software Development, Dynamic System Development Methodology (DSDM) and Kanban” (Sverrisdottir et al., 2014:258). Customer interaction has been examined in XP contexts (Grisham & Perry, 2005; Kautz, 2009). However, there are contrasting views on customer interaction in the utilisation of XP. Some studies point out the challenges of customer interaction in XP. For example, Grisham & Perry (2005) examined the relationship between customers and developers in software projects where XP has been adopted. They argue that it is not always feasible to have customers on site to perform frequent testing of software applications because this activity is not beneficial to the customer’s organisation, since the customer has to be situated at the developer site during the course of the project. Similarly, Koskela & Abrahamsson (2004) also pointed out that in small software projects where agile methods are used, customers’ time may be underutilised and as a result create a negative effect overall. Other studies point out the good aspect of customer interaction in XP. For example, Kautz (2009) presented the result of a case study conducted in a large software project that utilised XP. The study revealed that customer involvement in the project had a positive impact on the overall output and it enabled customers to perform active roles in software projects. Mohammadi et al. (2009) report both the positive and negative aspect of XP, as they point out that while there is productivity in software projects where customers are involved, there are also challenges associated with XP, such as customer unavailability and unnecessary customer changes in requirements.

Furthermore, Wang et al. (2008) identified the problems that may occur when customers and developers collaborate. These problems include customers not being available to participate in the development process, customers defining requirements not supported by the developers and insufficient customer skills for running test cases. Although the study examined the relationship between customers and developers, it did not explore the level of customer interaction in the Namibian context.

Customer interaction in Scrum projects is realised through the role of the product owner (Sverrisdottir et al., 2014). Sverrisdottir et al. (2014) point out that Scrum is not entirely practised, as organisations include other approaches to Scrum. However, they suggest that customers believed that incorporating Scrum into software projects improved the chances of successfully implementing software projects. Similarly, a study of agile teams in India and New Zealand by Hoda et al. (2011) reveals that even in agile projects, customers are not deeply involved in software development.

Besides customer interaction in Scrum software projects, Moe et al. (2010) suggest that customer interaction is not the only factor that influences a successful project; teamwork among software developers also plays a role. Hence, Moe et al. (2010) examined teamwork among developers in a Scrum project, using Dickson and Mcintyre's teamwork model. They deduced that management had to be involved to coordinate activities within a software development team. In agreement with Moe et al. (2010), Grapenthin et al. (2015) suggest that coordination among software developers is important in the software development process. Grapenthin et al. (2015) introduce an "interaction room" method in Scrum projects for software team members to improve mechanisms for identifying pending activities which need to be implemented before sprints are carried out. The approach seemed to be productive when applied in a software project.

Studies of the waterfall model reveal that waterfall model mainly emphasise customer involvement during the requirements-gathering phase of software projects (Hughey, 2009). Conversely, in agile methodologies, customers should be involved in most stages of the software development process (Bahta et al., 2012). The current literature implies that customer interaction is better achieved when agile software methodologies are applied to software projects. However, there is a limited number of current studies examining the differences in customer interaction among the different software methodologies specifically in the Namibian context. This study therefore addresses the gap identified in the current literature.

3. RESEARCH METHODOLOGY

Patton and Cochran (2002) state that qualitative methods are used when a researcher intends to answer the "what", "how" or "why" of a phenomenon. Since this study attempts to answer the research question "what are the differences in software methodologies deployed in Namibian software firms?" the qualitative method was adopted. The interpretive approach was used to meet the objectives of the study. This study also adopted a case study approach.

This study was conducted in three software firms in the Khomas region of Namibia. In line with the agreement between the researchers and the software companies to protect the identities of the participants, the names of these software companies were codified as Software_Firm_001, Software_Firm_002 and Software_Firm_003. These companies were purposefully selected to include only companies whose major service is "software development". Ten participants, consisting of four software project managers and six software developers, were purposefully selected from these software companies as shown in Table 1.

Table 1: Participants in this Study

Software firms	Software project managers	Software developers	Total
Software_Firm_001	2	2	4
Software_Firm_002	1	2	3
Software_Firm_003	1	2	3
Total	n=4	n=6	n=10

Turner (2010) states that standardised, open-ended interviews allow participants to provide comprehensive answers to interview questions. In an effort to provide comprehensive answers to the research question, standardised open-ended interview questions were used as data collection instruments. Each participant was interviewed separately. Each interview session lasted approximately for thirty minutes and was recorded and subsequently transcribed. Notes were also taken during the interview.

4. Results

The findings are categorised and presented under the following headings: software methodologies deployed; performance and management; challenges and costs. The findings are presented in subsequent sections.

4.1 Software Methodologies Deployed

Participants from each software firm were asked to identify the different software methodologies deployed in their software firms. All participants from Software_Firm_001 indicated that they had deployed the waterfall model and Scrum. A software developer from Software_Firm_001 commented: "Normally, we use the waterfall model, but the Scrum methodology was introduced to us, so we have adopted it recently in our projects". A software project manager from Software_Firm_001 added that "the waterfall model had always been our style, but since we decided to improve our processes, we have applied Scrum". All participants from Software_Firm_002 indicated that they had deployed the waterfall model, iterative model, XP, Scrum and RAD. A software project manager from Software_Firm_002 stated: "For a long time we have been using the waterfall model, but in recent years we have adopted other methodologies. As a result of the shortcomings of waterfall, we switched to other methodologies like the iterative model, extreme programming, Scrum; at a point we used RAD". All participants from Software_Firm_003 indicated that they had deployed the waterfall model, XP and Scrum. A software developer from Software_Firm_003 stated: "We use a variety of software methodologies when developing software applications, we have used waterfall, Scrum, extreme programming; we select methodologies based on the type of project and some other reasons."

4.2 Performance and Management

Software project managers identified precise planning at the beginning of software projects when the waterfall model was adopted. A software project manager from Software_Firm_001 indicated that the "time schedule in waterfall methodology can be easily planned because when I have all the requirements at the beginning of the project, I know exactly how long each activity should take and I can easily design a time schedule for my team." In Scrum-related projects, it was perceived that the use of user stories and customers prioritising activities helped to improve the software development process. A software developer from Software_Firm_003 indicated: "We know what we are expected to do first when the customer places priorities on activities". Scrum is also perceived to be a software methodology that enhances coordination between software firms and customers' companies, as a single person who is familiar with the business processes acts as a customer representative who interacts with the software team. As explained by a software project manager from Software_Firm_002, "We don't have to deal with all the customers or a group of customers; rather we deal with one customer who does the communication from both ends, that's quite easy to manage". Participants from Software_Firm_002 believed that the customer interaction in the iterative model improved the overall performance of software development, as no strict requirements were necessary in the early phases. Participants from Software_Firm_002 also indicated that with the iterative model, customers had the opportunities to refine changes as needed till the end of the project. Specifically, a software developer from Software_Firm_002 stated: "The good aspect of the iterative model is that we don't explicitly define the requirements at the beginning, but customers give their inputs later on, it also helps us as developers because we only develop the aspects the customers want." Another software developer form Software_Firm_002 stated:

"The iterative model is flexible, it allows changes and we don't have any expectations at the beginning, customer interaction in refining the software process makes it a flexible approach."

The majority of the participants who had adopted XP in their software firms admitted that the effect on customer interaction on the software development process is that it reduced the workload of software developers. A project manager from Software_Firm_003 stated: ''There is a huge difference, we feel more comfortable using the extreme programming approach, it has less documentation procedures like the waterfall model. With our conversations with the customers, we concentrate on the software development itself. I noticed that my team does not like documenting and the extreme programming approach tries to reduce this workload for them."

Participants from Software_Firm_002 believed that customer interactions in RAD facilitate a conducive work environment as formal processes do not have to be followed while customers continuously give feedback and suggestions. A software developer from Software_Firm_002 stated: "Software development with RAD is much easier; the constant interaction with customers doesn't require you to follow strict formal procedures".

4.3 Challenges

Time delays in the deployment of software applications were reported in the waterfall model, iterative, Scrum, XP and RAD. Customer interactions at any stage of the software development process are time-consuming, as changes are often requested for all software methodologies identified by software developers. Time delays in the deployment of software applications were attributed to constant changes in customer requirements. One project manager from Software_Firm_003 indicated: "Software projects are usually not deployed at the time you initially thought it would be deployed, because customers want to make changes here and there." A software developer from Software_Firm_002 indicated: "Well, there has not been any time when customers don't request for change, even if they were present at all stages of the project and this can slow down the completion of the project".

Customer-developer arguments were a common concern among software developers and software project managers. One software project manager from Software_Firm_001 stated: "It becomes complicated when the software developer and customer are saying two different things. This happens mostly in the waterfall model, customers have minimum interaction in the development process, they are usually brought in during requirements gathering and testing." A software project manager from Software_Firm_003 indicated: "There are times software developers don't understand the point being made by the customer and this could lead to verbal outbursts."

In the interviews regarding customer interaction in XP, challenges were related to frequent customer changes in requirements that could change the entire scope of the project. A software developer from Software_Firm_002 stated: "When you constantly interact with the customer or the customer is always there with you, it can also be challenging because they will always ask for changes and at the end of the day you have a completely different software application."

It was also gathered from the interviews that customer interaction in XP and RAD were challenging as it often affects planning. A software project manager from Software_Firm_002 commented: "Project completion time is not always easy to estimate at the beginning of the extreme programming and RAD projects, customers tend to change requirements most of the time and you really can't estimate what those changes would be at the beginning of the project". Another software developer from Software_Firm_003 stated: "XP can be demanding when it comes to customers refining the software, sometimes they don't know what they want, every time they add new things, making frequent changes like that affects your work".

4.4 Customer Satisfaction

Participants perceived that customer satisfaction was difficult to achieve through the waterfall model, as the majority of the participants indicated that customers were available during the requirements-gathering phase and during the user acceptance testing phase. A software developer from Software_Firm_001 stated: "Honestly, waterfall is difficult when it comes to giving customers what they want. Customer interaction is mostly done in the requirements-gathering phase after the development; the customer is called to test the application. Sometimes the client still requests changes they did not initially request for, despite them being involved in the requirements gathering process." It was often highlighted by participants that Scrum, XP, the iterative model and RAD achieved greater levels of customer satisfaction because requirements were adapted as soon as the request was made. A software developer from Software_Firm_003 stated: "XP is more focused on the customer need, it tries to incorporate what the users want and at the end of the day, customers get what they want." A software project manager from Software_Firm_002 stated: "The iterative model is flexible and at the end of the software development process, you are sure that customer has exactly what is needed". A software developer from Software_Firm_002 stated: "To achieve customer satisfaction, I would recommend extreme programming or Scrum and even RAD, the level of customer interaction is so good that despite the challenges, the customer is happy with what you have developed."

4.5 Costs

The findings from the software firms that participated in the study indicated that the level of customer interaction in the waterfall model increased costs to both customers and software firms. A software project manager from Software_Firm_001 stated: "When we follow the waterfall model, we do the requirements first, sign off the requirements stage and move on to the other stage of the project, we found out that this is a problem because at the later stage of the project, customers always want to change the scope or change the initial requirements. This was quite challenging because it was a waste of time and resources on our side since we invested so much time in the initial requirement-gathering phase and the development phase, so we charged the customers for this process and sometimes they were not happy." Participants indicated costs are much cheaper in software projects that explore XP, Scrum, the iterative model and RAD, as a result of the level of customer interaction allowed by these methodologies. A software project manager from Software_Firm_003 stated: "With the extreme programming approach, our customers have the opportunity to make changes to the initial prototype, as much as they want, without us spending too much time on the requirements-gathering phase, either way they are still going to change the requirements." A software developer from Software_Firm_002 stated: "RAD to me is much cheaper if you are looking at cost because customers always give their input and this reduces the chances of rework at the end of the project".

5. Discussion

This study investigated the software methodologies deployed in Namibian software firms as well as the differences in customer interaction with the software methodologies deployed in Namibian software firms. To the best of the researchers' knowledge, this is first time such a study was conducted within the Namibian context. This study also contributes to the body of literature that investigates customer interaction in software methodologies by empirically identifying the differences in customer interaction between software methodologies in Namibian software firms.

The findings revealed that software methodologies deployed in Namibian software firms include the waterfall model, Scrum, XP, the iterative model and RAD. The findings indicate that despite the uptake of different software methodologies, the waterfall model has been widely used in Namibian software firms. This is in line with other studies which indicate that the waterfall model is a widely used software methodology (Petersen et al., 2009; Munassar & Govardhan, 2010; McCormick, 2012). The findings revealed that customer interaction in the waterfall model, Scrum, the iterative model, XP and RAD cause delays in the deployment of software applications. This is in contrast with previous studies, which indicate that agile methodologies provide faster approaches to delivering software projects (Sharma et al., 2012). This may be attributed to the fact that the other methodologies, apart from the waterfall, might have been deployed recently in these software firms, taking a while to adopt. The findings revealed that software developers in Namibia preferred software methodologies with less formal procedures to software methodologies with formal procedures. Despite the challenges of the waterfall model, the findings also revealed that early planning was easier with the waterfall model. This is consistent with previous studies, which indicate that waterfall facilitates planning before software development (Kumiega & Van Vliet, 2008).

The challenges posed by customer interaction in software development in the waterfall model, Scrum, XP, the iterative model and RAD as highlighted by the software firms that took part in this study seem to be consistent with previous studies that reveal customer interaction may lead to additional challenges in the software development process (Mohammadi et al., 2009; Grisham & Perry, 2005). In addition, the findings from this study indicate that constant customer interaction could have a negative impact on planning in XP and RAD.

The findings also revealed that customer satisfaction is easier to achieve through agile methodologies, which is in line with previous studies which indicate that agile methodologies facilitate customer satisfaction (Kavitha & Thomas, 2011). Aside from time delays in the deployment of software applications, challenges were not highlighted for the iterative model and Scrum, in contrast to previous studies that indicate that getting the customer fully engaged in the software development process was a challenge in Scrum projects (Cho, 2008). It could mean that Namibian software firms are comfortable with the iterative model and Scrum.

The findings also indicate that cost was higher when the waterfall model was used. This is consistent with previous studies which indicate that cost is higher with the waterfall model (Balaji & Murugaiyan, 2012).

While other studies are congruent with this study that the waterfall model requires documentation (Balaji & Murugaiyan, 2012; McCormick, 2012), this study furthermore indicates that software developers in Namibia do not like to document. The differences in customer interaction are highlighted in Table 2.

Table 2: Summary of the Differences in Customer Interaction between Software Methodologies

Software methodologies identified	Performance and Management	Challenges	Customer Satisfaction	Costs
Waterfall methodology	Customer interaction in early stages enables initial planning	Customer interaction may lead to time delays in the software development process Customer interaction may also lead to customer-developer disagreement	Lower levels of customer satisfaction	Higher costs
Scrum	Customer interaction enables planning and coordination between customer company and software firms	Customer interaction may lead to time delays in the software development process	High levels of customer satisfaction	Lower costs
Extreme programming (XP)	Customer interaction reduces workload for software developers as less documentation is required	Customer interaction may lead to time delays in the software development process Constant customer interaction may lead to frequent changes in to software development Constant customer interaction may lead to planning problems as a result of the flexibility tolerated in this approach	High levels of customer satisfaction	Lower costs
Iterative model	Customer interaction improves overall performance of software applications as there is flexibility based on continuous customer interaction	Customer interaction may lead to time delays in the software development process	High levels of customer satisfaction	Lower costs
Rapid Application Development (RAD)	Customer interaction creates a conducive work environment for both customers and developers	Customer interaction may lead to time delays in the software development process Constant customer interaction may lead to planning problems as a result of the flexibility tolerated in this approach	High levels of customer satisfaction	Lower costs

In addition, the findings of this study reveal that customers are involved in software development processes, in contrast to previous studies in New Zealand and India where agile methods have been applied, and customers were not deeply involved in the software development process (Hoda et al., 2011).

6. Conclusions

The aim of the study was to identify the different software methodologies deployed in Namibian software firms and to examine the differences in customer interaction with the software methodologies deployed in Namibian software firms. The study provides useful insights in software methodologies deployed in Namibian software firms and highlights the differences in customer interaction with the different software methodologies. From this study, it was possible to establish that customer interaction, despite having a positive impact on the software development process, can also be challenging. The study discovered that customer interaction in agile methodologies does not always facilitate a faster approach to producing software. The study further found that customer interaction can affect planning.

From a managerial perspective, the findings suggest that customer interaction should be properly managed in software development projects to improve software development processes.

The sample size of three software firms used as case studies may have had an impact on the result. Furthermore, the selection of one region within the Namibian context constitutes a major limitation in generalising the findings. For future research, the sample size should be increased to generalise the findings.

7. References

Akinnuwesi, B.A., Uzoka, F.M., Olabiyisi, S.O., Omidiora, E.O. & Fiddi, P. (2013) An Empirical Analysis of End-User Participation in Software Development Projects in a Developing Country Context. *Electronic Journal of Information Systems in Developing Countries*, 58, 6, 1-25.

Anderson, R. & Mørch, A.I. (2013) Get Satisfaction: Customer Engagement in Collaborative Software Development. In Dittrich, Y., Burnett, M., A. Mørch, D. Redmiles (Eds.) End-User Development Springer Berlin Heidelberg, pp.235-240.

Awad, M.A. (2005) A Comparison between Agile and Traditional Software Development Methodologies. Honours Thesis, University of Western Australia.

Balaji, S. & Murugaiyan, M.S. (2012) Waterfall vs. V-Model vs. Agile: A Comparative Study on SDLC. *International Journal of Information Technology and Business Management*, 2, 1, 26-31.

Bassil, Y. (2012) A Simulation Model of the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology*, 2, 5, 1-7.

Bahta, B., Ikuh, H & Jama, A. (2012) Impacts of Effective Communication and Customer Collaboration on Information Development: A Case Study of Signma Kudos AB, Bachelors Thesis, University of Gothenburg.

Business Dictionary. (2014) Customer. http://www.businessdictionary.com/definition/customer.html

Chandra, V. (2015) Comparison between Various Software Development Methodologies. *International Journal of Computer Applications*, 131, 9, 7-10.

Cho, J. (2008) Issues and Challenges of Agile Software Development with SCRUM. *Issues in Information Systems*, 9, 2, 188-195.

Dingsøyr, T., Nurur, S., Balijepally V. & Moe, N.B. (2012) A Decade of Agile Methodologies: Towards Explaining Agile Software Development, *Journal of Systems and Software*, 85, 6, 1213-1221.

Dybå. T. & Dingsøyr, T. (2008) Empirical Studies of Agile Software Development: A Systematic Review, *Journal of Information and Software Technology*, 50, 9, 833-859.

Fogelstrom, N.D., Gorschek, T., Svahnberg, M. & Olsson, P. (2010) The Impact of Agile Principles on Market-Driven Software Product Development, *Journal of Software Maintenance and Evolution: Research and Practice*, 22, 1, 1-23.

Franken, S., Kolvenbach, S., Prinz, W., Alvertis, I. & Koussouris, S. (2015) CloudTeams: Bridging the Gap between Developers and Customers during Software Development Processes. *Procedia Computer Science*, 68, 188-195.

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.

Grisham, P.S. & Perry, D.E. (2005) Customer Relationships and Extreme Programming. *Proceedings of the 2005 Workshop on Human and Social Factors of Software Engineering*, 30, 4, 1-6.

Haikin, M. (2013) Reflections on Applying Iterative and Incremental Software Development Methodologies (Agile, RAD etc.) to Aid Development Work in Developing Countries. https://matthaikin.files.wordpress.com/2013/03/agile-blarticle-part-11.pdf

Hoda, R., Noble, J. & Marshall, S. (2011) The Impact of Inadequate Customer Collaboration on Self-Organizing Agile Teams. *Information and Software Technology*, 53, 5, 521-534.

Hoyer, W.D., Chandy, R., Dorotic, M., Krafft, M. & Singh, S.S. (2010) Consumer Cocreation in New Product Development, *Journal of Science Research*, 13, 3, 283-296.

Hughey, R. (2009) The Traditional Waterfall Approach http://www.umsl.edu/~hugheyd/is6840/waterfall.html

Humphreys, A., Grayson, K. (2008) The Intersecting Roles of Consumer and Producer: A Critical Perspective on Co-Production and Prosumption, *Sociology Compass*, 2, 3, 1–13.

Javanmard, M. & Alian, M. (2015) Comparison between Agile and Traditional Development Methodologies. *Cumhuriyet University Faculty of Science Journal*, 36, 3, 1386-1394.

Kautz, K. (2009) Customer and User Involvement in Agile Software Development. *Proceedings of the 10th International Conference*, 31, 168-173.

Kavitha, C.R. & Thomas, S.M. (2011) Requirements Gathering for Small Projects Using Agile Methods. *International Journal of Computer Applications Special Issue on Computational Science-New Dimensions & Perspectives (3) Published by Foundation of Computer Science*, 122-128.

Keil, M. & Carmel, E. (1995) Customer-developer Links in Software Development. *Communications of the ACM*, 38, 5, 33-44.

Koskela, J. & Abrahamsson, P. (2004) On-Site Customer in an XP Project: Empirical Results from a Case Study. *Proceedings of the 11th European Conference on Software Process Improvements*, 3821, 1-11.

Kumiega, A. & Van Vliet, B. (2008) A Software Development Methodology for Research and Prototyping in Financial Markets. *arXiv preprint* arXiv:0803.0162.

Lohan, G. Lang, M. & Conboy, K. (2011) Having a Customer Focus in Agile Software Development. In J. Pokomy, V. Repa, K. Richta, W. Wojtkowski, H. Linger, C. Barry, M. Lang (Eds.) *Information Systems Development*, Springer New York, Springer Science+Buisness Media, LLC, pp. 441-453.

Maddison, R., Baker, G., Bhabuta, L., Fitzgerald, G., Hindle, K., Song, J., Stokes, N & Wood, J. (1984) Feature Analysis of Five Information Systems Methodologies. In T. Bemelmans (Ed.) *Beyond Productivity: Information Systems Development for*

Organisational Effectiveness, Elsevier Science Publishers, North Holland Press, pp. 277-306.

McCormick, M. (2012). Waterfall vs. Agile Methodology. http://www.mccormickpcs.com/images/Waterfall_vs_Agile_Methodology.pdf

Moe, N. B., Dingsøyr, T & Dybå, T. (2010) A Teamwork Model for Understanding an Agile Team: A Case Study of a Scrum Project. *Information and Software Technology*, 52, 480-491.

Mohammadi, S., Nikkhahan, B. & Sohrabi, S. (2009) Challenges of User Involvement in Extreme Programming Projects, *International Journal of Software Engineers and Its Applications*, 3, 1, 19-32.

Moniruzzaman, A.B.M. & Hossain, S.A. (2013) Comparative Study on Agile Software Development Methodologies, *Global Journal of Computer Science and Technology*, 13, 7, 1-25.

Munassar, N.M.A. and Govardhan, A. (2011) Comparison between Traditional Approach and Object-Oriented Approach in Software Engineering Development. *International Journal of Advanced Computer Science and Applications*, 2, 6, 70-76.

Patton, M.Q. and Cochran, M. (2002) A Guide to using Qualitative Research Methodology http://fieldresearch.msf.org/msf/bitstream/10144/84230/1/Qualitative%20research%20methodology.pdf

Petersen, K., Wohlin, C. & Baca, D. (2009) The Waterfall Model in Large-Scale Development. In F. Bomarius, M. Oivo, P. Jaring, P. Abrahamson (Eds.) *Product-Focused Software Process Improvement*, Springer Berlin Heidelberg, pp.386-400.

Riley, J. (2012). Marketing who is the Customer. http://www.tutor2u.net/business/gcse/marketing_who_is_the_customer.html

Safwan, M.M.M., Thavarajah, G., Vijayarajah, N., Senduran, K. & Manawadu, C.D. (2013) An Empirical Study of Agile Software Development Methodologies: A Sri Lankan Perspective. *International Journal of Computer Applications*, 84, 8, 1-7.

Saiedian, H. & Dale, R. (2000) Requirements Engineering: Making the Connection between the Developer and the Customer, *Information and Software Technology*, 42, 419-428.

Sharma, S., Sarkar D. & Gupta, D. (2012) Agile Processes and Methodologies: A Conceptual Study. *International Journal on Computer Science and Engineering*. 4, 5, 892-989.

Sun, H., Yau, H.K. & Suen, E.K.M. (2010) The Simultaneous Impact of Supplier and Customer Involvement on New Product Performance, *Journal of Technology Management and Innovation*, 5, 4, 70-82.

Sverrisdottir, H.S., Ingason, H.T. & Jonasson, H.I. (2014) The Role of the Product Owner in Scrum-Comparison between Theory and Practices. *Procedia-Social and Behavioural Sciences*, 119, 257-267.

Tanner, M. & Willingh, U.V. (2014) Factors Leading to the Success and Failure of Agile Projects Implemented in Traditionally Waterfall Environments. *Proceedings of Management, Knowledge and Learning International*, 693-701

Terblanche, N.S. (2014) Some Theoretical Perspectives of Co-creation and Co-production of Value by Customers, *Acta Commercii*, 14, 2, 1-8.

Turner, D.W. (2010) Qualitative Interview Design: A Practical Guide for Novice Investigators, *The Qualitative Report*, 15, 3, 754-760.

Wattanakamolchai, S. (2008) Managing Customer Participation in the Service Production Process, *Boston University Academic Review*, 8, 1, 1–9. http://www.bu.ac.th/knowledgecenter/epaper/jan_june2009/pdf/Somyot.pdf

Wang, X., Wu, Z. & Zhao, M. (2008) The Relationship Between Customers and Developers in Agile Methodology, *International Conference on Computer Science and Information Technology*, 566-572.

Wattanakamolchai, S. (2008) Managing Customer Participation in the Service Production Process, *Boston University Academic Review*, 8, 1, 1–9.

Williams, L. & Cockburn, A. (2003) Agile Software Development: It's about Feedback and Change. *Computer*, 36, 6, 39-43.

Young, D. (2013). Software Development Methodologies. https://www.researchgate.net/profile/David_Young30