Documentation Practices for Offshore Agile Software Development

Faria Kanwal, Komal Bashir, Ayesha Haider Ali

Department of Computer Science, Lahore College for Women University, Lahore, Pakistan
faria.kanwal@gmail.com

Abstract: Agile software development (ASD) has gained a lot of momentum in the field of software development. Agile emphasize greatly on the customer satisfaction and close interaction with the development team. Software development has reformed with the increase in telecommunication facilities. Now the globally dispersed teams works on the software development. With the increase in offshore software development, the close interaction is not possible among the stakeholders of the software. The geographically dispersed stakeholders of the software may result in ambiguous and inconsistent requirements. Due to the cultural barriers and time zone limitations, communication and interaction on daily basis is not possible. The quality of software may suffer due to lack of interaction. Although, agile do not support heavy documentation, it is suggested to complement light-weight documentation to increase the quality and reduce the risk factor in geographically distributed environment. The current research mainly focuses on the necessary documentation to be maintained during the offshore agile development to enhance the quality, communication and to reduce the risk. This approach aims to help the stakeholders in development of the software.

[Faria Kanwal, Komal Bashir, Ayesha Haider Ali. **Documentation Practices for Offshore Agile Software Development.** *Life Sci J* 2014;11(10s):70-73]. (ISSN:1097-8135). http://www.lifesciencesite.com. 12

Keywords: Agile software development (ASD), Offshore Software Development (OSD), Outsourcing, Documentation Practices

1. Introduction

Agile Software Development (ASD) is a place of practices that is formed by the experienced personnel (Chow & Cao, 2008). According to the William and Cockburn, ASD is all about the continuous change in requirements and provide feedback; it is an iterative development approach in which multiple iterations are involved to develop software incrementally (Saleh, 2011). Agile focuses on the working software over heavy documentation reducing the development time as well. This methodology emphasizes on close interaction of the developer with the customer. The unavailability of customer may lead to software failure. (Khan, et al., 2011) states that software development outsourcing is a contract-based relationship between client and suppliers, (suppliers are the provider of the contracted desired/agreed). Offshore software development (OSD) is delivery of software development services by a supplier situated at a distant location from the one where the software will be used. The main reason behind companies using offshore software development services is the higher development cost of the local service providers. Offshore practices and issues are addressed from global point of view instead of customer supplier view point. In offshore development, the lack of trust between the parties may cause the failure. Close interaction is a major challenge in geographically distributed environment. (Krishna, et al., 2004) identifies that the most of the software development may lead to failure due to the lack of communication. This problem can be overcome by introducing the necessary documentation in agile environment where the stakeholders are distributed. This research focuses on the essential documentation in agile environment when the customer and development team is geographically dispersed. The aim of this paper is to provide guideline of necessary documentation for offshore ASD.

2. Literature review

2.1 Agile Software Development

The concept of agile came into existence in 2001 with the agile manifesto where certain conventional methods of software development were disregarded and new ones were established. In software industry, ASD compliments the primary flow with the number of practices known as agile manifesto for the ASD (Chow & Cao, 2008). ASD facilitates requirement change even in the later stages of the development (Bari & Ahamad, 2011; Frauke Paetsch, 2003). It aims to deliver high quality software and to satisfy the customer needs, customer satisfaction is given highest priority (Dyba & Dingsøyr, 2008; Hass, 2007). Customer is involved in the development stages of the software along with the developers from the feasibility to the deployment of the software. This was done on the basis of the failures of these conventional methods where the development process was too rigid to squeeze the changed requirements in the later stages of development. The lack of customer participation, in the development process, was another key point in its failure. Another perspective was that over indulgence in the documentation work, where each step taken forward, each meeting etc. had to be written down to maintain a record, led to ignoring the key factors of the software development. Thus development of software within the estimated time frame was difficult. Agile on the other hand had some key principles where customer satisfaction, team collaboration and their ability to handle changes in the development were much appreciated. ASD consist of small iterations; a development team is involved in all iterations throughout the software development i.e. planning, requirement gathering, requirements analysis, design and coding, implementation and testing a build (Kettunen, 2009). ASD has different flavors for example; Extreme Programming (XP), Scrum, Feature Driven Development, Crystal family.

Agile imposes certain limitations on the development lifecycle such as introducing agile methods in larger complex systems (Andrea & Qusef, 2010), availability of the customer with the developers for longer periods of time (Dingsøyr, 2009), effective communication among the team members if the team size is too large and lack of documentation or not having enough documentation (Andrea & Qusef, 2010; Hoda, et al., 2011).

2.2 Offshore Development

Offshore Development is distributed development environment, done in different continents across the globe. It is delivery of software development services by a supplier situated at a distant location from the one where the software will be used Offshore Outsourcing Development is catalyzed due to the globalization of information technology and the improvement of telecommunication facilities. (Nakatsu & Iacovou, 2009) in their study highlighted the offshore outsourcing disastrous due to the problematic organizational and cultural barriers, mismanagement, middle management resistance and communication failure.

Offshore development failure is usually misunderstood and miss handled due to poor understanding, no precautionary primitives, or considering the external factors affecting offshore development in future. IT projects threats and risks are also growing with boom span of software industry, delays (Khan, et al., 2011), resources overruns (Leffingwell, 1997), and poor requirement engineering (Felfernig, et al., 2010).

(Kliem, 2004) highlighted the risks to offshore development like import-export barrier, political instability, language barriers, cultural issues and currency exchange fluctuations which does not make the end product materialize the expected outcomes. (Krishna, et al., 2004) stated that every society has certain working standards, process and practices which are usually distinct in cross-border development. It also identifies cultural issues, relationship management, staffing and training utter to be a hurdle in effectiveness of the offshore development.

(Khan, et al., 2011) addresses the issues that intend to reveal in offshore development. Mismanagement of organizational, cultural, government or political barriers lead to failure of an outsoutced project. (Rajkumar & Mani, 2001) stated that the proper planning should be introduced, conducted and managed throughout the life cycle. Moreover, (Leffingwell, 1997) identified that end-user and offshore development team could not have face-to-face or informal meetings which results in inconsistent, incomplete and ambiguous requirements.

(Kliem, 2004) diagnosed the project communications language plays a vital role for formal or informal meetings. In offshore development language, accent, terminology and slangs are misunderstood due to cultural norms.

User involvement is a challenge in offshore development, low or no communication can cause delays, conflicts and rejections. In offshore client expectations are critical to meet because users don't communicate directly to developers (Krishna, et al., 2004).

3. Documented approach of agile in offshore development

Due to the close interaction of customer and team, agile do not support extensive and detailed documentation. The problem arise when modifications required and the team is geographically dispersed. In this paper, the focus of the study is towards the importance of documentation in the agile environment if the software is outsourced. Since agile follows lesser documentation than traditional approaches (Kanwal, et al., 2010), the outsourced software faces the quality consequences. Existing methods for improving documentation will be explained and then a proposed solution will be provided keeping this background information as a baseline.

(Rubin & Rubin, 2011) investigated that the documentation during the software development not only helps during the project completion but can be used for knowledge sharing when the development team becomes unavailable. Project artifact like change decisions, requirement elicitation, and customer feedback are the enough documents that can be managed in a project to overwhelm any inconvenience during the software development (Hoda, et al., 2012). A worthy documents profoundly describes a software system, its components and relationships among different components (Stettina & Heijstek, 2011). It is valuable to better understand the system and communicate among the team members. Lesson learned during the project helps in knowledge sharing, transfer, evolution and maintenance (Uikey, et al., 2011). (Hadar, et al., 2013) also describes the importance of abstract specification of the project for evaluation, assessment and communicate in agile environment. The above mentioned problem can be overcome by introducing some necessary documents along with the recommended steps. Proposed essential documents are feasibility report, software requirement specification, software design document, validation document and lessons learnt.

3.1 Feasibility report

Feasibility report consists of a brief introduction of project along with the technical details like hardware and software feasibility. The technical details contains the requirements of the hardware, software and any additional software required to operate the system successfully. Moreover, this section also describe the schedule, budget and resources.

3.2 Software requirement specification

Software requirement specification document contains the functional and non-functional requirements. Functional requirement helps to identify the required functionality of the system that is traceable at the time of verification and validation of the system (Akram, et al., 2014). Requirement specification can be managed in the sequence diagrams that is easily understandable for the geographically dispersed teams.

3.3 Software design document

The design phase can be divided into two broad categories; high level design and detailed design. The two system design helps to effectively communicate the dispersed teams in terms of functionality of the system. Requirement specification document along with the design document not only helps in the development of existing system, it also improves the system maintenance. Shared visualization of these document collaborate and communicate the geographically distributed teams in an efficient and effective manner.

3.4 Validation document

Agile development highly emphasize the customer satisfaction. Validation process deals with the customer satisfaction in which the series of testing is done on the basis of the customer specifications. Integration, system and user acceptance testing is recorded to meet the goals of the customer.

3.5 Lessons learnt

OSD and ASD have changed the meaning of traditional development of the software. It encompasses the close collaboration of customer with the highest satisfaction level but on the dispersed locations. In this regard, lessons learned from the system development play a vital role to address the development of similar nature projects when the development teams is dispersed.

4. Conclusion

Documentation is an important part of software development. Though agile do not significantly emphasize on heavy documentation but can accommodate the light-weight documentation. The geographically dispersed stakeholders of the software may communicate through the documents. This documentation approach results in high quality software with the reduced risk of the software development. The documented approach may help in the maintenance of the software.

Corresponding Author
Ms. Faria Kanwal
Computer Science Department, Lahore College for Women University, Lahore, Pakistan
E-mail: faria.kanwal@gmail.com

References

1. Akram, M. U. et al., 2014. A Study on RE Process Models for Offshore Software Development. *Journal of Basic and Applied Scientific Research,* 4(4), pp. 114-119.
2. Andrea, D. L. & Qusef, A., 2010. Requirements Engineering in Agile Software Development. *Journal of Emerging Technologies in Web Intelligence.*
3. Bari, M. A. & Ahamad, S., 2011. Managing Knowledge in Development of Agile Software. *International Journal of Advanced Computer Science and Applications,* 2(4), pp. 72-76.
4. Chow, T. & Cao, D.-B., 2008. A survey study of critical success factors in agile software projects. *The Journal of Systems and Software,* Volume 81, pp. 961-997.
5. Dingsøyr, T. D. a. T., 2009. What Do We Know about Agile Software Development. *IEEE.*
6. Dyba, T. & Dingsøyr, T., 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology 50 (2008) 833–859,* Volume 50, pp. 833-859.

7. Eran Rubin, H. R., 2010. Supporting agile software development through active documentation. *Springer.*
8. Felfernig, A. et al., 2010. Recommendation and decision technologies for requirements engineering. *In Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering (RSSE '10),* pp. 11-15.
9. Frauke Paetsch, D. A. E. D. F. M., 2003. Requirements Engineering and Agile Software Development. *IEEE.*
10. Hadar, I., S.Sherman, E.Hadar & J.J.Harrison, 2013. *Less is more: Architecture documentation for agile development.* San Francisco, CA, IEEE, pp. 121-124.
11. Hass, K. B., 2007. The Blending of Traditional and Agile Project Management. *PM World Today.*
12. Hoda, R., Noble, J. & Marshall, S., 2011. The impact of inadequate customer collaboration on self-organizing Agile teams. *Information and Software Technology,* Volume 53, pp. 521-534.
13. Hoda, R., Noble, J. & Marshall, S., 2012. Documentation strategies on agile software development projects. *International Journal of Agile and Extreme Software Development,* 1(1), pp. 23-37.
14. Kanwal, F., Bashir, K. & Fahiem, M. A., 2010. *A Hybrid Software Architecture Evaluation Method for FDD – An Agile Process model.* s.l., IEEE.
15. Kettunen, P., 2009. Adopting key lessons from agile manufacturing to agile software product development—A comparative study. *Technovation,* Volume 29, pp. 408-422.
16. Khan, ". U., Niazi, M. & Ahmad, R., 2011. "Barriers in the selection of offshore software development outsourcing vendors: An exploratory study using a systematic literature review. *Information and Software Technology,* pp. 693-706.
17. Kliem, R., 2004. Managing the Risks of Offshore it Development Projects. *EDPACS,* 32(4), pp. 12-20.
18. Krishna, S., Sahay, S. & Walsham, G., 2004. Managing cross-cultural issues in global software outsourcing. *Commun. ACM,* apr, 47(4), pp. 62-66.
19. Leffingwell, D., 1997. *American Programmer,* 10(4), pp. 13-16.
20. Nakatsu, R. T. & Iacovou, C. L., 2009. A comparative study of important risk factors involved in offshore and domestic. *ELSEVIER, Information & Management,* p. 57–68.
21. Rajkumar, T. & Mani, R., 2001. Offshore Software Development The View From Indian Suppliers. *Information System Management,* 18(2), pp. 1-12.
22. Rashina Hoda, J. N. a. S. M., 2012. Documentation strategies on agile software development projects.
23. Rubin, E. & Rubin, H., 2011. Supporting agile software development through active documentation. *Requirements Engineering,* 16(2), pp. 117-132.
24. Saleh, M. F., 2011. An Agile Software Development Framework. *International Journal of Software Engineering (2011): 97-106.,* pp. 97-106.
25. Stettina, C. J. & Heijstek, W., 2011. Necessary and Neglected? An Empirical Study of Internal Documentation in Agile Software Development Teams. *ACM.*
26. Uikey, N., Suman, U. & Ramani, A., 2011. A Documented Approach in Agile Software Development. *International Journal of Software Engineering.*

5/27/2014