

Generalized Framework for Agile Software Development Process

Bhalerao Shilpa¹, Ingle Maya².

¹Acropolis Institute of Technology and Research, Indore, (M.P.) INDIA

bhalerao.shilpa@gmail.com

²Indore Institute of Computer Application, Indore, (M.P.) INDIA

maya_ingle@rediffmail.com

Abstract- Agile Software Development Process (ASDP) has been adopted by many software industries. There exist various methods based on ASDP. Agile methods (AMs) are used mostly in small and low critical projects due to lack of generalized practices. Limited applicability of AMs in distributed environment, life critical system, contract projects and large complex application generated the strong need of generalized framework for ASDP. In this paper, we have proposed a Generalized Agile Software Development Process Framework (GASDPF) for ASDP. It covers four aspects of ASDP mainly; redefining ASDP, a set of ASDP practices, identifying the suitability of ASDP in particular organization and environment and usefulness of ASDP. GASDPF is a step towards building trust among the practitioners.

I. INTRODUCTION

Agile Methods (AMs) being capable to accommodate volatile requirements in high tight schedules [1]. And hence, these are adopted by Google, Microsoft, Yahoo, IBM and partially by others also. There exist various methods based on Agile Software Development Process (ASDP) viz. Extreme Programming (XP), Scrum, Feature Driven Development (FDD), Crystal Methodology (CM), Dynamic Software Driven Method (DSDM), Adaptive Software Development (ASD) etc. and provide higher customer satisfaction, low defect rates, higher usability and a solution to higher changing requirements [2][3]. Also, these methods offer practices for improving productivity and quality of software [4]. As AMs are people centric, the success of these methods largely depends on team management. Applicability of AMs is well suited for small projects with low life critical system [5]. In practice, XP is mainly used with Scrum due to unavailability of management practices [6][7]. AMs also emphasize on experienced and knowledgeable team members due to the collaborative work culture. Therefore, average developers in team may find difficulty to understand and manage entire agile approach for development [5] [6]. However, there is a need to develop a framework for ASDP to provide the guidelines to team members so as resolve the aforesaid issues.

In this paper, we propose a framework for ASDP that incorporates definition, suitability, usage and implementation of agile software development. Necessity of agile framework is discussed in Section 2.

We propose a Generalized Agile Software Development Process Framework (GASDPF) in Section 3. In Section 4, we conclude with future scope.

II. NECESSITY OF FRAMEWORK

In this section, we will discuss the major issues that recommend the necessity of ASDP framework.

It has been observed that XP possesses many development practices such as Test Driven Development (TDD), pair programming, re-factoring etc. for implementing the principles of agile manifesto. At the same time, there is lack of evidences about use of development practices in other methods such as ASD, Scrum etc. [7]. Moreover, Scrum is enriched with management practices whereas other methods such as FDD, ASD and XP etc. possess very few project management practices. It has been found that XP is using maximum existing agile practices as shown in Fig. 1. However, all these methods bring flexibility by accepting the requirements, applying short iterations, simple design and small releases but there is lack of some important practices in all categories (development, testing and management). Many evidences have been found about failure of AMs due to reluctant management and unavailability of environment and culture [9]. Suitability of ASDP is always a topic of debate due to organizational and technical issues. Nevertheless, implementing agile practices with tradition software development causes misunderstanding about agile principles. Thus, there is scope for assessment of suitability of ASDP in particular project and in environment before applying AMs in project. Therefore, limited applicability of AMs in distributed environment, life critical system, contract projects and large complex application generated the strong need of framework to cope with these limitations [11].

III. GENERALIZED AGILE SOFTWARE DEVELOPMENT PROCESS FRAMEWORK (GASDPF)

This framework is a conceptual foundation that brings good practices together to enrich the development approach and includes set of principles, models, discipline, concepts and guidelines for designing application according to ASDP. It covers four aspects of ASDP mainly; redefining ASDP, a set of ASDP

REVIEW PAPER

International Journal of Recent Trends in Engineering, Vol 2, No. 4, November 2009

practices, identifying the suitability of ASDP in particular organization and environment and usefulness of ASDP as shown in Fig. 2 and are discussed as follows:

A. Redefining ASDP

It intends to build the theoretical foundation for ASDP and synthesizes the theory of software development and practices used in it by providing the complete characteristics of ASDP. We have proposed a new definition of ASDP stated as:

“An incremental, iterative and disciplined light weight approach for software development to achieve maximum level of customer satisfaction through customer collaboration, responding to change, delivering working software and emphasizing individual interaction which is especially beneficial for particularly disruptive software development.”

B. Practices

we have classified agile practices in three categories namely; development practices, testing and quality practices and project management practices and discuss them in detail as follows:

a. Development Practices

Existing primary development practices according to GASDPF are shown in Table 1 whereas new primary practices include version control, risk analysis, early build, reusability and reengineering practices.

Version control keeps all project artifacts in a single, authoritative place and facilitates simple tracking and monitoring the changes in software. Risk analysis must be performed to mitigate the risk thereby increasing return on investment as well as build trust and confidence of team. It is performed at beginning of iteration for deciding prioritization of requirements. GASDPF includes TDD, pair programming and re-factoring as optional development practices as average developers may not able to use concept of these practices judiciously. Hence, GASDPF recommends the use these practices in presence of experts to avoid inconsistency in data and code. Task board defines the tasks to be performed by the end of day, backlog tasks of previous day as shown in Table 1. The use of reengineering practices, usability practices have been recommended in case of legacy system to reduce time and efforts required to develop the system.

b. Testing and quality improvement Practices

The primary testing practices of ASDP are namely; code inspection, customer acceptance test, automated testing, integration testing and UI testing and code review whereas new practices introduced are tracking issue and code change notification as shown in Table 2. Tracking issue keeps details of product such as versions, issues, details of notifying customer, level of severity, frequency, fix time and names of team members (who fix the issue) and verification of issue. Another, interesting practice

introduced in GASDPF is code change notification to all stakeholders thereby keeping everyone informed about change. GASDPF recommends new optional testing techniques for quality assurance of the developed software namely; smoke test, exploratory test and mock client test as shown in Table 2. Smoke testing is a kind of integration testing and is performed on common usage scenarios involving changing components. Exploratory testing discovers emergent behavior, unexpected defects, and risks related to quality attributes such as security, performance and reliability. Mock client test is performed in case of unavailability of customer for ensuring the highest customer satisfaction.

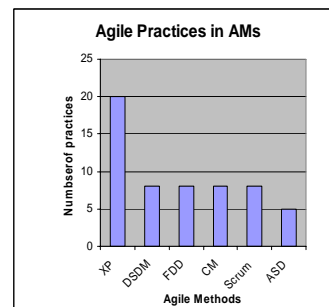


Fig. 1 Agile Practices in AMs

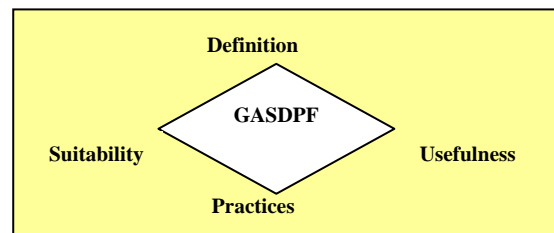


Fig. 2 GASDPF Framework

TABLE 1 GASDPF Development Practices

Primary Practices	Optional Practices
Coding Standard	Test Driven Development
Configuration Management	UI Refactoring
Continuous Code Integration	Pair programming
Continuous Database Integration	Task board
Flexible Architecture	Usability evaluation practices
Simple incremental Design	Reengineering practices
Version Control	
Early build	
Risk analysis	
Data naming convention	

TABLE 2 GASDPF Testing and quality Improvement Practices

Primary Practices	Optional Practices
Code Inspection	Code Refactoring
Customer Acceptance Test	Database Refactoring
Automated unit testing	Smoke test
Integration Testing	Exploratory testing
UI Testing	Mock client test
Code review	
Track Issues	
Code change notification	

TABLE 3 GASDPF Project Management Practices

Primary	Optional
Active Stakeholder participation	Regular Status report
Requirement or feature prioritization	
Active team member communication	
Iteration planning	
Iterative development	
Incremental Delivery of working Software	
Self organized and discipline team	
Small releases	
Velocity tracking	
Early and iterative Estimation	
Root cause Analysis	
Retrospective	

c. Project Management Practices

GASDPF includes existing project management practices such as stakeholder participation, requirement prioritization, iteration planning, incremental delivery of working software and small releases as listed in Table 3. Early and iterative estimates are primary practices used to improve the estimates of the project whereas velocity tracking is useful to identify the efforts made by team in developing new requirements along with error correction of previous working software. Root cause analysis prevents reoccurrence defects by identifying the reasons. Retrospective is the way of continuously improving the process and product by analyzing the process and product at end of iteration. Regular status report is an optional practice that determines the status of project at any stage of software development life cycle.

C. Suitability of ASDP

Three most important success factors of ASDP mainly are culture, people and communication. Organization culture must be changed from command-and-control to leadership-and-collaboration. ASDP is people centric process and defines people as team members, customer and user for the development. People must encourage new ideas, knowledge sharing, pair programming, reflection workshops and collaborative decisions. Effective communication plays a vital role to transfer the information among the team and customer. ASDP emphasizes on face-to-face communication for reducing uncertainty in requirements.

D. Usefulness of Agile Process

GASDPF defines usefulness of ASDP for customer and development team. Many practitioners advocate ASDP due to organizational, technical and personal successes. Agile team increases the value of product by including business experts and developing the core product that deliver financial and operational returns to the organization. Prioritization of requirements play important role in organizational success by keeping most valuable feature at the highest priority. AMs advocate the use advanced tools such as project management tools, automated testing tools etc. that lead to technical

excellence. ASDP practitioners have claimed that it also improves personal skills as agile practices encourages individual interactions, technical debates etc. Agile practitioners enhanced their skill set due to use of practices such as retrospective, iteration planning and root cause analysis.

IV. CONCLUSION

Our proposed framework, GASDPF is a step towards building trust among the practitioners. The advantages of using this framework in software development are:

- 1) It provides transparent understanding of ASDP to average managers and developers.
- 2) The framework clearly identifies the scope of the project with respect to ASDP by examining the issues related to management, people, process and technology.
- 3) It also indicates that AMs can be used in life critical system as it involves practices of short iteration, simple design and rigors testing with proper study of inception and feasibility.
- 4) GASDPF provides the guidelines to use ASDP in legacy projects by introducing new development practices, testing and management practices.
- 5) Framework defines project management practices to track and monitor the progress as well as utilization of resources by the means of status report, early estimation and iteration planning practices. These practices eliminate the need of merging the practices of AMs with traditional methods.
- 6) This framework will establish a benchmark in classification of agile and non-agile practices.

REFERENCES

[1]Clegg, C. et al., "The Performance of Information Technology and Role of Human and Organizational factors", *Economic and social research council 1996*.
 [2]Lycett M., Macrede R., Patel C., Paul R., "Migrating Agile Methods to Standard Development Practice", *IEEE Computer June 2003 pp. 79-85*.
 [3]Abrahamsson P., Salo O., Ronkainen J. and Warsta J., "New Directions on Agile Methods : A Comparative Analysis", *Proceeding 25th International conference on Software Engineering 2003*.
 [4]Robinson H. and Sharp H., "XP Culture: Why the Twelve Practices both are and are not most Significant Thing", *Proceedings of Agile Development Conference 2003*.
 [5]Boehm B., "Get Ready for Agile Methods, with Care", *Computer vol. 35, no. 1 2002, pp. 64-69*.
 [6]Cockburn A., *Agile Software Development Series, Pearson Publication low price edition*.
 [7]Agile Adoption Survey at <http://www.surveymonkey.asp?SID=3463544&Rnd=7.819378>
 [8]Agile Manifesto at <http://www.agilemanifesto.org>
 [9]Turk D. and Rumpe B., "Limitation of Agile Software", *Proceedings of International Conference on eXtreme Programming and Agile Process in Software Engineering, 2002*.
 [10] Parsons D, Ryu H., and Lal R., "Impact of Methods and Techniques on Outcomes of Agile Software Development Projects", *Proceedings of International Federation for Information Processing, 2007, pp. 11-26*.