



International Journal of Engineering, Business and Enterprise Applications (IJEBA)

www.iasir.net

Modified Agile Process: Improving the Competency for Process Development Methodologies

Sanjeev Dhawan^a, Divyendu Kumar Mishra^b

^aFaculty of Computer Science & Engineering, ^bResearch Scholar

Department of Computer Science & Engineering, University Institute of Engineering and Technology,
Kurukshetra University, Kurukshetra-136 119, Haryana, INDIA.

E-mail: rsdhawan@rediffmail.com, ratn01mishra@gmail.com

Abstract: Agile methodologies are gaining popularity in industry although they compromise a mix of accepted and controversial software engineering practices. Both global software development and agile approaches have gained significant popularity. Companies even show interest in applying agile approaches in distributed development to combine the advantages of both approaches. This is done despite their differences in key tenets. In their most radical forms, agile and global software development can be placed in each end of a plan-based/Agile spectrum because of how work is coordinated. In this paper, an attempt has been made to discuss some of the modifications in Agile process that will improve the efficiency of existing Agile Processes.

Keywords: Code Refactoring, Sprint, Sprint Planning, Sprint Tracking.

I. Introduction

Software engineering is a discipline devoted, as all engineering, to solve problems from our daily lives. Specifically, software engineering is concerned with all aspects of software production. Even though software construction was not considered an engineering process for a long time, it has proved to possess all the properties, risks, and requirements to be expected from an engineering activity [1].

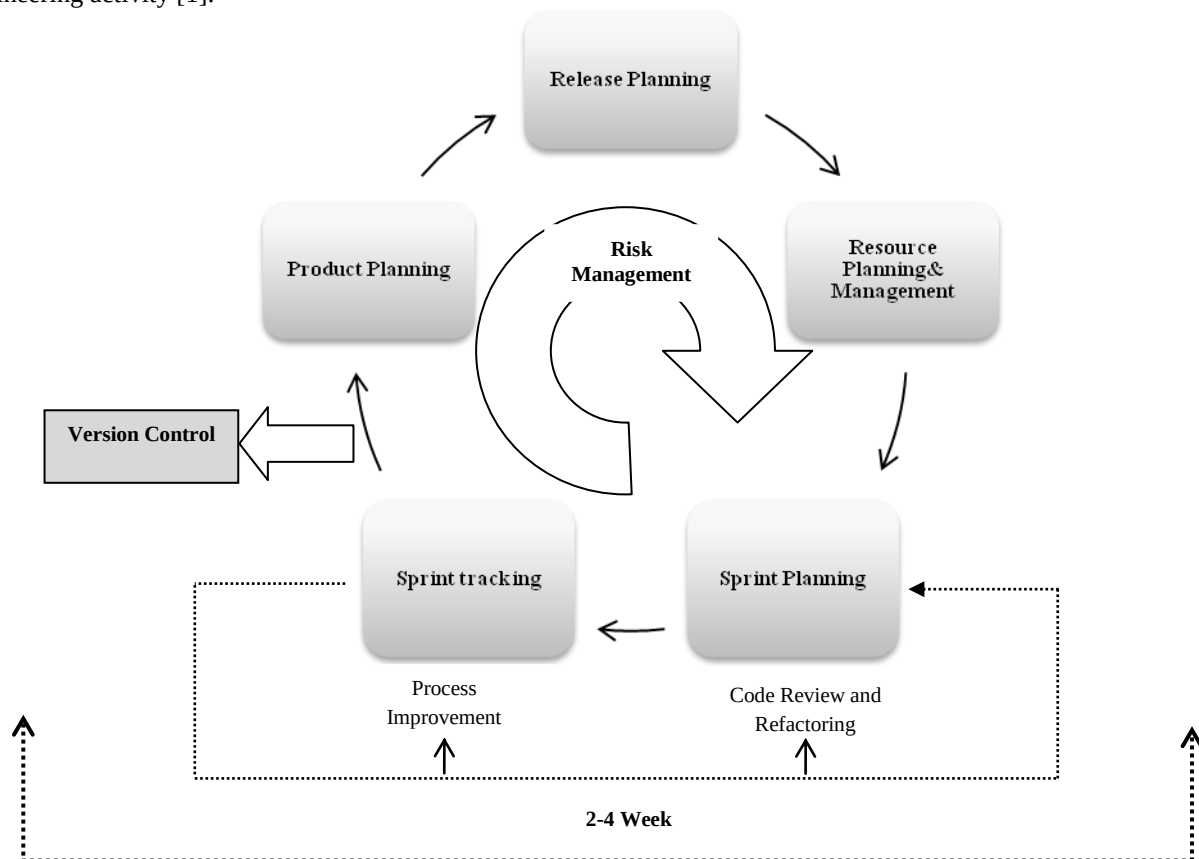


Figure: Modified Agile Process

The “Modified Agile process” for distributed projects is an approach to global delivery model in Agile way. To fulfill the Agile manifesto, “Modified Agile process” has incorporated many of its best practices from different Agile processes. “Modified Agile process” involves the risk management in the entire process and version control after the completion of one cycle. It preserves the framework and prevents the violation of architecture. “Modified Agile process” support process improvement and code review and refactoring. In this model process improvement and code-Refactoring provide the facility to change a source code without modifying its external functional behavior. According to Gartner [2], globalization of software development has expanded rapidly in recent years and has brought in its wake changes that impact application development projects. Global Software Development (GSD) is becoming the norm for many technology companies. According to Lindstrom and Jeffries [3], the traditional popular project management techniques focus on developing a plan and sticking to the plan. This improves coordination but reduces the ability of the project to adapt to new information regarding requirements or implementation details. However, traditional project management techniques do not take into account that the customer will be in the US and the development teams will be in India and China. The problems brought by distance are not taken into consideration. Moreover, they cannot meet with the dynamic requirement of the GSD.

II. Project Initialization

During project initialization phase of “Modified Agile Process” requirements from the customer/stakeholders are captured through the project manager who is an initial contact person for customers. At the very first initial project meeting several activities are conducted: project planning and scheduling, allocation of resources, development strategies including development plan, configuration management, and development of frame base. In this initial phase of the project, stakeholders and the project management team prioritize the feature lists. Project manager, software architect, test and quality manager, feature designer, configuration manager and executive programmers are involved in this phase. Modified Agile Process aims to offshore project information sharing and cross site education. Modified Agile Process implements two or three features from the priority list in each cycle which is duration of 2-3 weeks. During this short cycle detailed analysis, design, implementation, test are done for each feature. Quick prototype of the implemented features is delivered for customer evaluation at the end of each cycle.

III. Life cycle of Modified Agile Process

There are five phases in the Modified Agile Process named as Product Planning, Releases Planning, Resource Planning and Management, Sprint Planning and Sprint Tracking. Along these phases Risk Management runs parallel with each phase. After the completion of one cycle when new cycle is going to be start then Version Control activity must also be done for the purpose of maintaining unique version of each application. In between the phase of sprint planning and sprint tracking it uses Process Improvement and Code Review and Refactoring.

- (i) **Product Planning:** Product Planning allows the Customer to define, organize and prepare the project team's backlog. New stories are created to define new functionality to be added to the system. These stories can be organized into higher level groupings and broken down into smaller components more suitable for delivery.
- (ii) **Release Planning:** The primary goal of release planning is scheduling items into releases. At this same point in the planning cycle, however, other important activities may occur.
- (iii) **Resource Planning and Management:** In organizations adopting Agile, Resource Planning and Management is a common challenge when they want to allocate people among projects. There is a tendency to think of projects as the primary unit of organization. In this view, the project is static and we move people and resources to the project in order to staff it. When the project is finished, we return the people and resources to the central pool and make them available for the next project. Some projects demand the use of specialists who are in limited supply. In those cases, you may need to share specialists across many teams and the teams will need to adjust their Agile schedules around the availability of the specialists. Finally, the biggest problem in most organizations is too many projects and too little focus. Just like having too many cars on the highway, having too many projects in flight is a guarantee for slow delivery. There is plenty of inventory management and queuing theory to support this and we all know it intuitively. Most IT organizations have a lot of projects going on, hence a lot of spending going on, but not a lot of delivery happening. Each project (or car) is inching slowly along, fighting for attention and limited resources. The solution of course is take some cars off of the highway so that the remaining cars can fly! But this requires the political strength to say "I'm sorry but we are fully allocated and we cannot support your project right now. We will queue it up for the next available team."
- (iv) **Sprint Planning:** The sprint represents the basic timeframe for an Agile team. Within a given sprint, the team is fully delivering a small slice of the overall feature set. That includes doing all of the design, development, testing and documentation of the features. In this planning process, the focus shifts from the customer to the other team members as the primary drivers. This process typically takes place in one or two sessions at the beginning of the sprint with the entire team in attendance. Sprints are grouped into a sequential group called a schedule.

- (v) **Sprint Tracking:** Once the sprint is planned out by the team, they turn their attention to execution. As team members work on items, optionally track time, and record progress.
- (vi) **Version Control:** If we have several Agile development teams working on the same codebase, then it is necessary to minimize the risk of stumbling over each other. To ensure that there always is a clean, releasable version at the end of each iteration we should apply version control at the end of each iteration.
- (vii) **Process Improvement:** In the advance phase the flow of the project is evaluated. It allows the team to improve and customize the process according to their current needs. As well the team tracks all steps of development and uses its observations for waste reduction.
- (viii) **Code Review and Refactoring:** Re-factoring is the activity which includes changing a source code without modifying its external functional behavior. The intention of this activity is to improve some of the non-functional attributes of the software. As an outcome, team gets improved code readability and internal architecture which results in improved maintainability of the source code and extensibility.

IV. Project Responsibilities

Due to the large variations in sizes of the development teams the composition of the project roles can vary. The most common case is that the same person can represent several project roles. In Modified Agile Process, it proclaims that one shall keep the number of roles to a minimum level. Unlike SCRUM, Modified Agile Process needs a specific role definition. The core idea of the approach is that each individual takes responsibility for a given activity. It also suggests that the responsible subject gives the project management team, time estimations for their assigned activities. Following set of roles involved in the process:

- **Project manager:** This is responsible for monitoring the deliveries and to handle resources. The project manager shall also arrange feedback meetings with the stakeholders. As well this person is involved in the establishment of the time plan, being in communication with both the development team and the customer.
- **Architect:** An architect is responsible for establishment of one or more architectural rules for software providing good performance, testability, scalability, maintainability and modularity. An architect shall also interact with the test & quality manager when it comes to questions concerning the testability. Moreover, the architects' most important task is to form a strategy, which prevents the executive programmers to violate the architectural rules.
- **The Test Designer role:** The role of test designer is to defining the test approach, division of tasks related to testing and ensuring its successful implementation. The role involves identifying the appropriate methodologies, tools for implementation of the required tests.
- **The Feature Designer:** The feature designer is responsible for giving the very specific task of conceptualizing and designing a component (or a group of components) for a system. Normally, this includes features, modes, or other parts of software. They normally report to the architect.
- **Test and Quality Manager:** The test and quality manager incorporates a key role, with responsibility for the quality planning process and monitoring the implementation of the relevant quality standards. Another meaning of the position of test manager is this is the person who ensures the timely delivery of all testing activities and associated data in accordance with the prescribed quality standards.
- **Executive programmers:** These are responsible for development of the software, code writing. As well their responsibility includes unit test writing and performing.
- **The Configuration Manager:** It is responsible for configuration of the system and environment to the product development team. It supports the product development activity so that developers and testers have appropriate workspaces to build and their code. The configuration manager role also has to investigate and analyze product review, and change and defect tracking activities.
- **Delivery manager:** It is responsible for insurance that the deliveries are stable and don't contain any "show stoppers".
- **Requirements Manager:** The role of requirement manager is to elicit, analyze, specify, and validate the requirements received from the customer.
- **Risk Manager:** It is responsible for evaluation and minimization of the risks taken by the team during the development.
- **Stakeholder:** It is an integral part of a project. He is the end-users or client, the person from whom requirements will be drawn, the person who will influence the design and, ultimately, the person who will reap the benefits of completed project.

V. Conclusions

The goal of this research paper was to systemize and describe the existing stage of "Agile Process for Distributed Projects" and propose suggestions for its modification by the name of "Modified Agile Process". The first objective of the research was to systemize the practices, phases and other attributes of the process and to describe Modified Agile Process as precisely as possible. The second objective of this research was to identify issues causing negative effect on the process. Comparison with

other software development processes and analysis of the project experience revealed several leaks and weak points of the process. The third objective was actual improvement of the process. According to the detected weak points several improvements were proposed. They took different forms such as extra roles on the process, additional practices or activities which were not performed (or performed on insufficient level before).

VI. References

- [1] J. E. Tomayko and O. Hazzan, Human Aspects of Software Engineering, Charles River Media, Inc., 2004.
- [2] Iyengar,P.Application Development Is More Global than Ever, Publication G00124025, Gartner, 2004;
www.gartner.com/resources/124000/124025/application_dev.pdf
- [3] Lindstrom,L and Jeffries, R.Extreme Programming And Agile Software Development Methodologies, *Information Systems Management*, 24:3 , pp. 41-60, 2004
- [4] Cem Kaner, James Bach, and Bret Pettichord 2001. Lessons Learned in Software Testing. John Wiley & Sons, Inc.
- [5] Brian White and Geoff Glemm 2000. Software Configuration Management Strategies and Rational ClearCase: A Practical Introduction. Addison-Wesley Longman.