

# Configuration Issues and Efforts for Configuring Agile Approaches-Situational based Method Engineering

Rinky Dwivedi  
Computer Engineering Department  
Delhi Technological University  
Delhi, INDIA

## ABSTRACT

In software development process, the requirements are not stable and keep on changing according to the customers. In such dynamic situations the developers are required to use dynamic approaches for software development. The agile methodology provides dynamic methods such as extreme programming, scrum, feature driven approach and dynamic system development method. Studies reveals agile methods are not anti-methods and can be tailored as any other traditional methods. This paper investigates tailoring of various agile methods, approaches required to tailor the method components of agile methods. Further, this paper proposes the configuration issues and configuration efforts which must be kept in mind during tailoring of the agile components.

## Keywords

Agile methodology, dynamic approaches, configurability in software development process and situation method engineering

## 1. INTRODUCTION

Agile methodology means cutting down the big picture into small pieces and then bringing them together at the right time after the due process is over [1, 2, 4]. If we consider traditional method like Rational Unified Processes (RUP) as the league leaders then definitely agile methods are the cutting edge competitor to it. Although there are reasons to support both the traditional (waterfall) and agile methods, let's take a closer look to explore why many software and web design firms make the more appropriate choice of employing agile methodology.

Agile methodology supports the dynamic requirements of the customer. Customer here participates throughout the development process, gives their inputs and conducts tests for the desired output. In traditional software development customer involvement is just at the starting requirement phase and at the end testing phase. In agile methodology the requirements evolves during the whole development process in contrast to the fixed requirements in case of traditional methods. An integrated testing process is followed in the agile methodology throughout the development process, whereas in traditional software development method, testing is a phase that comes at the end of project development life cycle. Continuous customer involvement leads to the continuous feedback in case of agile and in case of traditional methodology feedback can be get only when the project or iteration is released and that again depend upon the software process model used.

Apart from the comparative analysis done above an industrial case study [3] shows that the agile methods offer low requirements volatility in projects and reduction of waste

(discarded requirements) in the requirements engineering process

This paper is laid out as follows. In the next section, issues with agile methodology are discussed, followed by section 3 devoted to the most popular agile methodologies i.e. XP and scrum. In section 4 various approaches for method tailoring is presented along with the approach for agile methods. Finally, the configuration issues and Configuration efforts in agile approaches are presented in section 5 and section 6 of this paper.

## 2. ISSUES FACED WITH AGILE METHODOLOGY

The agile methods are currently preferred over the traditional methods this new methodology brings following issues which are to be kept in mind during development of applications [4, 5].

A) Agile is difficult to implement as it requires continuous customer involvement which leads to more management responsibilities. The management has to be more open and take active participation in development process.

B) In agile approaches, the code is integrated after each iteration and agile has to be more disciplined to ensure application is always in a workable state.

C) In agile, for each stage planning has to be done and it has to be updated as well to meet the changing requirements, therefore strong planning must be done to develop projects using agile approaches.

D) The off-shore development is gaining popularity in last few decades, but in agile methodology it is still a challenge [23] as agile is centered on face to face communication. It supports an open space where the team can work closely together

Apart from these there are several new issues that are going to be addressed later in this paper in respect of the configuration management and tailoring of agile methods.

## 3. POPULAR AGILE APPROACHES

Presently number of agile software development methods are used in practice, some of them are; Extreme programming [6, 7], scrum [8, 9], dynamic system development method [10] and feature driven development [11]. A complete comparison or description of all the agile processes would be beyond the scope of this paper because they are too many. Therefore in this paper we present a brief description of the two most popular agile software development techniques i.e. Extreme programming and Scrum

### 3.1 Extreme programming (XP)

XP is described by Beck [7, 12] as a light weight methodology suitable for small-to-medium sized dynamic projects and supports good for the technical and coding aspects of development of project development life cycle. The life cycle of XP consists of five phases i.e. exploration, planning, iteration to release, productionising, maintenance and death.

**A).Exploration phase:** In exploration phase the project finalizes its tools and techniques required to develop the project on the other side customer starts writing the story cards. The stories written on the story card by the customer describes what the customer wants to include in the first iteration. The exploration phase takes time of few weeks to few months.

**B).Planning phase:** In Planning phase the schedule is prepared for the first and subsequent releases. The programmer first estimates the amount of effort required for each story and then prepare schedule accordingly.

**C).Iteration to release:** Before the first release the system has to go for several iterations as per the schedule set in the planning phase. The first iteration creates a system with the architecture of the whole system and at the end of each iteration; functional tests already created by the customer are used to check the system.

**D).Productionising phase:** In Productionising phase the decision has to be made for the newly found changes whether to include them in this iteration or not, if they exists. Extra testing and checking of the performance is done before actual delivery of the project to the customer.

**E).Maintenance phase:** Maintenance phase in XP projects requires a lot of support from customer's site. The project team has to keep track of both i.e. producing new iterations and system under running.

**F).Death phase:** This comes when system satisfies the customer requirements and customer does no longer have any stories to be implemented. Here necessary documentation for the system is finally written as no further changes to the architecture, design and code are made.

#### Limitations of XP

- XP is a code-centered approach rather than a design-centered approach; it may not matter in small projects, but becomes a serious limitation in larger projects (more than few thousand lines).
- Lack of quality plan- as compared to formal methods XP does not explicitly plan, measure or manage program quality.

### 3.2 Scrum

It is basically a framework, instead of describing detailed description of how everything is to be done; it lefts several issues on the shoulder of software development team. Scrum supports software development team and believes that the team will know best how to solve the problem they are presented. Scrum relies on self organizing and cross functional teams. The development team are supported by two main actors Scrum master and Product owner. Scrum master helps the team members use the scrum framework and Product owner guides the team towards building the right product. The life cycle of

scrum includes three phases: Pre-game, Development and Post-game.

**A).Pre game:** - The Pre game phase includes two sub phases: Planning and Architecture. Planning is the requirements gathering phase of scrum process. Here requirements can originate from all the actors involved in the system. These requirements are then prioritized and accordingly definition of the system is freezes. In Architecture phase, the architecture of the system is designed on the basis of definition developed in the planning phase. Design review meetings reviews the proposals made for the implantation and based on their review decisions are made.

**B).The Development phase:** - This phase is the agile part of the scrum process and is also called as Game phase. In the development phase the system is developed in Sprints. Sprints are the iterative cycles where the functionality of the system is enhanced by controlling different environmental and technical variables. In contrast with the traditional development system scrum aims at controlling these variables constantly throughout the process.

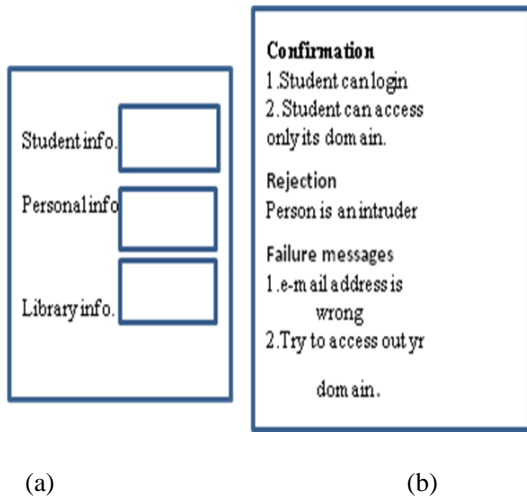
**C).Post Game:** - In Post Game phase, system is now ready for the release, system enters in this phase when an agreement has been made, that the environmental variables and technical variable are completely addressed. This phase includes the task such as integration, system testing and documentation.

#### Limitations of scrum

- In case a task is not well defined, accurate estimation of the project costs and time may not be forecasted precisely. In such a case, the task can be spread over several sprints.
- The success of Scrum projects depends upon the Scrum Master to a great extent. If they exercise strict control over the team members making them extremely frustrated, it may lead to failure of the project.

### 3.3 Examples of the user stories written during agile process

Agile process depends on writing good user stories with just enough details. For example if the developer epic is to create a new screen to input student information. The screen should have few subsections such as personal information, departmental information, library information etc. also the student should be able to add, update or simply view his records. So, the question comes in the developer mind is how to break the epic into good user stories? And how the developer can select the level to break? The interesting part of agile is it all depends upon the matter of opinion; however user's perspective must also keep in mind before actually writing the user stories. Agile gives a broad idea "try to break them down small, but not so small that they start to become heavily inter-dependent".



**Fig1 (a): Student user story card (front)**  
**Fig1 (b): Student user story card (back)**

#### 4. METHOD TAILORING

Fitzgerald [14] noted a recent recognition that “off-the-shelf” methods need to be tailored first to fit the needs of a specific project, even if the method appears to be appropriate and suitable for the project in hand.

“Situational Method Engineering (SME) is a solution offered to the problem of the selection of the “most appropriate” methodology for an organization and/or its projects.” [15]

In order to efficiently meet the current needs, methods need to be adapted to specific characteristics or project situations which lead to the development of situational method engineering (SME). A situation specific method is applicable only in one or more situations for which it was specifically developed. The concept of SME believes in retrieving, adapting and tailoring the method components rather than adopting complete methodology to specific situations. Once the components are retrieved [16] to construct situation specific method from method base, these components need to be tailored by assembly, extension or reduction.

There are several approaches to method engineering [15]

**Reduction-based:** Removing non essential concepts in an existing methodology

**Extension-based:** Enhancing an existing methodology With new concepts and properties;

**Assembly-based:** Constructing the methodology through Assembling method fragments retrieved from a repository.

In assembly [17], separate components are grouped together as per the need of a specific project. Whereas extension [17] transforms a base method into a new by adding some components and reduction [18], removes some components from the base method to match the user’s requirements. Assembly technique relies on a method base, on the other hand extension and reduction believes on a base method and evolving that method to meet user’s requirement.

#### Method tailoring in agile methods

An important feature of researches carried out on traditional formal software systems is that these are largely theoretical in nature and based on conceptual arguments to support how method should be tailored. Very little approaches are available

in light of practical applications, whereas agile methodology is completely dependent on the practical experience of the developer and requirements of the customer.

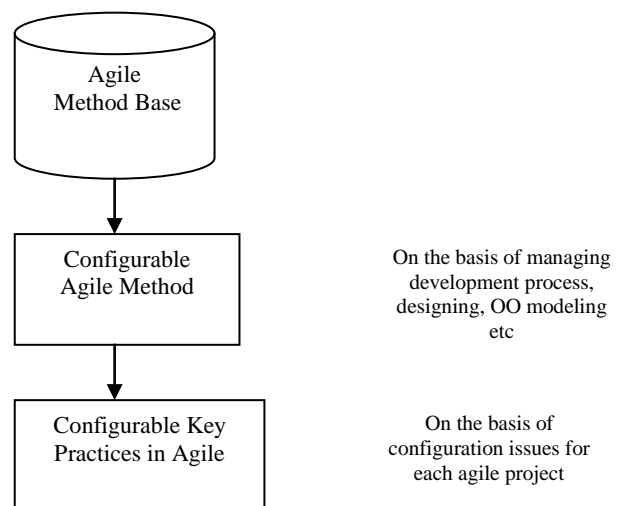
Since agile methods [19] are incremental they support small software releases, cooperative customer and developer working constantly together with close communication, straightforward and adaptive. To tailor the method components of agile, an ad-hoc approach is required. To support our view we discussed a brief view of the case study presented by Fitzgerald [20].

Developers at Intel Shannon studied agile and found that agile may individually be incomplete to support the development process well but their processes can be tailored. In this case study they have shown that XP is tailored and only 6 out of 12 key practices of it are used and combined with other agile practice i.e. Scrum. XP is particularly useful for technical development stages and Scrum provides the necessary overall project management process. So this shows that the agile methods can be extensively tailored to meet the specific need of the development context.

Agility which has two basic meanings flexibility and configurability [21] has become a very important characteristic in modern software development process. The agile software development process always focuses on the client and moves forward based on client’s recommendation. These processes are more people-oriented than process-oriented.

#### 5. CONFIGURATION ISSUES

Various agile methods are residing in the agile method base each of is having its own characteristics and is suitable for one situation or another .like from the details of the key practices and by the practical experiences of many developers it can be said [19] that XP is best suited for technical aspects and scrum provides support for project planning and tracking



**Fig 2: Configuring Agile Methods.**

While using any of the agile approach, continuous improvement, tuning and adjusting of the software development process is required. This results in several issues that come on the way of the configuration process in agile methodology. After studying and analyzing different agile processes and their key practices, this paper proposes following issues which must be kept in mind during the configuration process:

**A).Organizational Issues:** Configuration process adopts essential key practices and makes a choice between non essential

key practices. It depends upon the organization that how it chooses the non essential key practices. The requirement may vary during the different stages and developers finds it difficult to cope up with the varying organizational requirements. For example in XP there are two key practices - 'small release and coding standards'. The former is solely depends upon the silicon available and latter depends upon how early an organization defines its coding standards.

**B).Developers Issues:** Implementing a key practice and eliminating the other would also depend upon the developers of the project. Developers may or may not understand the requirement of the customer and end up convincing him / she on the variables to be configured depending on his own understanding, which may sometime proves to be fatal for the project. Also the developers among themselves are susceptible to commit errors while following the key practices. For example pair programming and 40 hours a week are two major key practices in XP .The success of these practices depends upon the mutual understanding between developers and collaboration between the developers those are working across time.

**C).Customer's Issues:** Customer always wants its project delivered as early as possible with utmost satisfaction to him / her. While doing to he / she may become adamant on inclusion of some key practices which may prove to be hurdles in the successful implementation of the project. This sometime also leads to some key practices to be left unused. In a special case like overseas project practice of 'on-site customer' becomes difficult to implement.

**D).Situational Issues:** - The configuration process also depends on the situation, specific to which the project development is undertaken. The situation in question may or may not permit inclusion or exclusion of the non essential key practices. For example if the projects in hand are more critical and need more protective measures to protect the project from intruders a key practice like collective code ownership cannot be used.

## 6. CONFIGURATION EFFORT

The first effort required in the configuration process is to identify all the stakeholders of the project then the agile process and its concepts to be used are explored. Such concepts are then prioritized as per the requirement of the organization and the processes are configured as per the usefulness and feasibility of the key practices used during the course of development process. The process so configured is then reevaluated to see that the goals of the development process are achieved or not before moving ahead with the overall project development.

$Ea$ =Essentials of New Method

$Ea'$ =Essentials of old method

$NEa'$ =Non-Essentials of Old Method

$Ma$ =New Method

$Ea' = Ea$

Means, Essential concepts of new method is equal to the Essential Concepts of old method

$NEa' \cup Ea' = Ma$

Non-essential concepts or variables of  $Ea'$  union essential of  $Ea'$ =New Method.

So, equation of effort is

*Effort to engineer method X from method Y is directly proportional to the number of non-essential concepts of X.*

## 7. CONCLUSION AND FUTURE WORK

This paper begins with exhaustive literature review and proceeds with in agreement with the theory that agile methods are not anti-methods and can be tailored as any other traditional formal methods. Our research reveals that the organization, Developers, Customers and situational issues should be given due consideration. Also a considerable effort is required compared to the traditional development method in using agile approaches. However by looking at the sizeable and qualitative advantages the agile approaches have an edge over the traditional approaches. Prima facie it may appear that the traditional approaches are almost free of issues and requires bare minimum efforts in the development process, but the agile approaches are still better and advantageous with increased satisfaction to the customers, if the issues are understood properly and sincere efforts are made during the development process. The future works will explicit focus on a project with identification of issues and efforts with the help of a case study.

## 8. REFERENCES

- [1] Chau, T.; Maurer, F.; Melnik, G.(2003). Knowledge sharing: agile methods vs. Tayloristic methods, Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings.
- [2] Johnson, M. (2002). Agile methodologies: Survey results. Victoria, Australia: Shine Technologies.
- [3] K. Petersen and C. Wohlin, "A Comparison of Issues and Advantages in Agile and Incremental Development between State of the Art and an Industrial Case", Journal of Systems and Software, Vol. 82, No. 9, pp. 1479-1490, 2009.
- [4] Abrahamsson P., Warsta J., Siponen M., Ronkainen J. "New directions on agile methods: a comparative analysis", Proceedings of the 25th International Conference on Software Engineering, 2003.
- [5] Nerur S., Mahapatra R., Mangalaraj G., "Challenges of migrating to agile methodologies", Communications of the ACM - Adaptive complex enterprises, Volume 48 Issue 5, May 2005.
- [6] Beck, K. (1999a). Embracing change with Extreme programming. IEEE computer 32(10):70-77.
- [7] Beck, K. (1999b). Extreme programming explained: Embrace change Reading, Mass., Addison-wesley.
- [8] Schwaber K., (1995) "scrum development process" OOPSLA'95 Workshop on Business Object Design and Implementation. Springer-Verlog.
- [9] Schwaber K., Beedle M.(2002). Agile software development with scrum. Upper saddle River, NJ, Prentice hall.
- [10] Stapleton, J.(1997). Dynamic system development method- The method in practice. Addison Wesley.
- [11] Palmer S.R., Felsing J.M., (2002). A practical guide to Feature Driven Development. Upper Saddle River, NJ, Prentice-Hall.
- [12] B. Bahli and E.-S. Abou-Zeid." The role of knowledge creation in adopting xp programming model: An empirical study". In ITI 3rd International Conference on Information

- and Communications Technology: Enabling Technologies for the New Knowledge Society, 2005.
- [13] Example of a user story *by* kelly waters, 18 january 2008 |agile testing, agile user stories.
- [14] Fitzgerald, B., Russo, N.L., O’Kane, T.: Software Development Method Tailoring At Motorola, Communications of the ACM 46(4), 2003, 65-70.
- [15] Deneckere R., Kornyshova E., Claudepierre B., Contextualization of Method Components, Proc. RCIS, 235-246, 2010.
- [16] Nehan Y. R. & Deneckère R. (2007). Component-based situational methods - A framework for understanding SME. In proceedings of the International IFIP WG8.1 Conference ME’07, Springer, Geneva, Switzerland.
- [17] Ralyté J., Deneckere R. & Rolland C. (2003). Towards a generic model for situational method engineering. In proceedings of the international conference CAISE’03, Springer Verlag, Velden, Austria.
- [18] Wistrand, K., Karlsson, F.: Method Components – Rationale Revealed. In Advanced Information Systems Engineering 16th International Conference, CAiSE 2004, Riga, Latvia, June 7-11, 2004, Proceedings, A. Persson, J. Stirna, Eds. Springer-Verlag, LNCS 3084, wBerlin, 2004, 189-201.
- [19] Abrahamson P., Salo O., Ronkainen J., ”Agile software development methods-Review and analysis”, Juhani Warsta, University of Oulu, 2002.
- [20] Fitzgerald B, Hartnett G and Conboy K (2006) Customising agile methods to software practices at Intel Shannon, European Journal of Information Systems, 15(2), pp. 197-210.
- [21] P Lou, Z Zhou, YP Chen “Study on the multi –agent-based agile supply chain management”, the international journal of advanced ..., 2004 – Springer.
- [22] Fowler M. “Using an Agile Software Process with Offshore Development “<http://www.martinfowler.com/articles/>, 2003.