HYBRID SOFTWARE DEVELOPMENT APPROACH FOR SMALL TO MEDIUM SCALE PROJECTS: RUP, XP & SCRUM

M. Salman Bashir
Department of Computer Science, University of Engineering & Technology, Lahore, Pakistan
salman.vu@gmail.com
M. Rizwan Jameel Qureshi
Faculty of Computing and Information Technology, King Abdulaziz University,
Jeddah, Kingdom of Saudi Arabia
anriz@hotmail.com
Cell # (+966-536474921)

ABSTRACT: Software industry is using various customized models such as Rational Unified Process (RUP), eXtreme Programming (XP) and SCRUM. XP and Scrum are most widely practiced and documented agile models. Both XP and Scrum work well for small projects whereas RUP is suitable for large projects. A fine integration of these models is required to deal with small, medium and large projects. This paper is written to highlight some of the characteristics, strengths and weaknesses of RUP, XP and SCRUM. In this research, a novel framework is proposed. The proposed framework is a fine combination of strengths of RUP, XP and SCRUM in order to achieve high quality software and enhance the team productivity.

Key words: RUP, XP, Scrum, Process Model, SDLC

1. INTRODUCTION

Document driven approaches are widely used now days in software industry due to its methodical and straightforward nature for all kinds of projects. These approaches are also providing better predictability and high assurance due to its potential benefits for large scale projects [1]. Document-driven approaches for software development provide a very structured and formalized way for stable requirements. That’s why these approaches are also very beneficial when we deal with large scale projects due to their straightforwardness. Rational Unified Process is also one of them, follows an iterative and incremental approach for software development [2]. Some of the conventional models include Waterfall Model, RAD Models, Spiral Model and RUP Model etc. [3]. The newly proposed novel framework contains RUP that is used not only to provide skeleton, structured and formalized way throughout SDLC but also strengthen XP practices through its philosophy. This novel framework will also highlight some of very useful practices from RUP, but their applicability depends on the nature and type of project.

Rational Unified Process is an incremental and Iterative, architectural centric approach, which is based on sound software engineering principles. It is a well defined process model that provides step by step guidelines to develop object oriented software [3]. RUP has been evolved [4] in different eras in different situations. RUP framework is of two dimensional one is called Phases and the other one is Disciplines. Phases in fact represent four major stages Inception, Elaboration, Construction and Transition [3]. Any application that follows RUP should have to gone through these four stages otherwise the project success could be at risk. Second dimension called Disciplines. These are the logical activities that need to be performed throughout the life cycle of the project. The disciplines are further divided into sub-disciplines; one is called main discipline while the other one is support Discipline [5].

Agile is a term used for a collection of software development techniques that gain popularity in the beginning of 21st century. Unlike traditional waterfall approaches, these techniques emphasize on improved customer satisfaction, adapting to changing requirements, frequently delivering working software, and close collaboration of business people and developers [6]. It has been highlighted through studies and surveys that unlike Document Driven companies, the companies which opted agile approach are more customer-centric and satisfy the customer more than document driven companies [3]. One reason for this observation is the ability of agile approach to welcome change in requirements rather discouraging it as in the traditional approach. Hence Agile helps creating a more satisfactory relationship with the client [7].

Scrum is a popular agile technique used to develop and manage software. It is necessary to understand certain roles and responsibilities of people in Scrum lifecycle. There is a **Product Owner** whose responsibility is to define business value and requirements of project. Moreover the Product Owner also prioritizes the requirements. Another stakeholder in Scrum is the **Scrum Master**. Scrum Master is the Project Manager of the project [8]. Extreme Programming is the most famous agile technique. XP use story cards for elicitation. A user story is the description that provides business value to the customer. Below is a diagram that depicts the life cycle of Extreme Programming approach. Moreover the techniques like brainstorming, interviews and prioritization are also used in elicitation process. In XP, there are certain rules that need to be followed [3]. Extreme Programming provides best engineering practices for a good quality product at small scale [9].

Section 2 describes literature review. Section 3 illustrates research design. Section 4 covers the proposed framework.

2. RELATED WORK

RUP is a document driven, heavy weight approach for software development and is still widely acceptable in software industry for its numerous and tremendous strengths. RUP provides a very structured and methodical way of development for large scale projects [2][3].

RUP gives high predictability, stability and good quality results for large scale projects due to its straightforwardness [10]. RUP focuses significantly on producing more documentation whose major benefit is to manage and maintain the system rather than on tacit knowledge of individuals [5].

RUP guides and helps software developers and testers through defined testing mechanism for complex and large scale projects. RUP is very suitable in case of stable requirements [10]. RUP mainly focus on process rather than individual people, [11]. That's the reason changing in development team structure or leaving any individuals doesn't affect too much to overall success. RUP provides a well defined mechanism to model the business as compared to agile. RUP defines and provides an established way to handle non-functional requirements as compared to agile [12].

RUP has the capability to manage the whole development efficiently with new technology as compared to agile [13]. RUP provides a good mechanism and formalized way to capture user requirements through different elicitation techniques as compared to agile. RUP provides a well defined procedure for software deployment as compared to agile. RUP provides and establish a defined link from use cases to code as compared to agile [3][4][5].

RUP is unable to manage and handle thee frequently changing business requirements and that's the reason, RUP has the tendency to be over budgeted and behind the schedule specially in changing requirements [1]. RUP also doesn't produce very good results to provide reliability, simplicity and to achieve more improvements in productivity in changing business requirements [14].

It is experimentally proved by different researchers that during software development, Requirements often get changed by at least 25% or more [15]. RUP as being a conventional method was not initially designed to cope with these changing business and technical requirements efficiently. So, it's also one of the limitations of RUP to make complete set of requirements upfront. Delivered application's functionality is too little when we compare it with spent time [16]. Sometimes, it is seen that the built application is more than desired [17] and built lower priority first.

Developed application is of poor quality due to high software bugs rates. It s also seen that poor project management results in assigning wrong resources that leads to project failure [18].

Good Requirement Engineering and Management is one of the key factors in the success of the projects. According to the Standish Group Survey, incomplete requirements and changing requirements and specifications were significant contributors to project failure [19]. In another study, Taylor reports that "70% of projects failed at requirement definition stage and 80% also claimed that clear and detailed requirements were a critical success criterion." [19]. One of the most important aspects of requirement management is managing the changing requirements. Requirements get changed during the course of development process as well as after the software releases are shipped to the customer. There are many reasons for requirement changes. Among several reasons an important reason is the lack of communication and coordination between the customer and the development team. Therefore unlike Waterfall or traditional development approach, Agile has devised a technique to keep continuous relationship with client.

In order to understand how agile produce good results as compared to RUP in changing business requirements, at first, we need to know the reasons for changing requirements. There are several reasons for requirements to change. Some of the important reasons are described as follows: [20].

- Customers and end users usually do not have complete understanding of their needs and requirements. But as the project progresses they start developing better understanding of the domain and hence request for changing the already stated requirements.
- As the development proceeds, some inconsistencies in requirements are highlighted. This leads to requirement change.
- During the period of development or in the maintenance stage, the priorities of the customer get changed and hence changes in requirements are proposed.
- The requirements can change if the platform or the environment of the system changes due to some reason.
- Requirements can also be changed if it is too expensive or too difficult to implement under the set of given conditions.

The changes in organization for which the software is being developed can also result in changing the business requirements. RUP is an object oriented process model that takes an incremental and iterative approach and is based on sound software engineering principles [12]. It describes well defined disciplines that acts as a skeleton for all kinds of projects especially for large scale but doesn't give simplicity, dramatic change in progress, frequent adaptation to changing requests [21]. Scrum provides best mechanism to manage and track the application development but lacks of structured approach and best engineering practices [18]. XP has best engineering practices but lack of structured and formalized nature as well as lack of best managerial aspects [22].

Our core objective/problem is to develop high quality software and to increase the productivity by combining the strengths of XP, Scrum and RUP models that is much more beneficial for clients. In order to achieve the described goal, it is the need of the hour to combine the strengths RUP, Scrum and XP while suppressing their weaknesses. Project Managers face biggest challenge in exercising suitable methodology for their projects. Selecting a suitable methodology enables to bring quality in a project and completes it in less time and according to schedule.

3. RESEARCH DESIGN

The primary intention of conducting this case study is to develop an application that is based on the guidelines of Hybrid Proposed model. There were total six team members and the duration of this case study was 1.75 months. Online CIIT Library Management System was developed and total number of iteration required to build this system was four. There was a training session before the starting of application development with an objective to get much more understanding about Hybrid Proposed model.

Team members were excellent in their skills while performing term applications. They also had satisfactory information about agile methodologies and their respective practices. They also had good knowledge about RUP process model before moving towards implementation. However, in order to achieve better quality results there was a training session about the Proposed Hybrid model description in which they were completely familiarized about Scrum management procedures, roles and artifacts. They were also given a complete description about very useful XP engineering practices and basic philosophy like simple design, pair programming, test driven approach, continuous integration, coding standards, usefulness about the on-site availability of customer as well as different quality and productivity aspects of XP practices. They were also given detail knowledge about RUP process models, logical activities and their objectives as well as all related tasks required to perform in each logical activity. In short they were given a complete set of steps how to proceed in application development. Team of developers, testers and others worked under the guidelines of Scrum master. It was first ensuring that the team structure must balance in each aspect.

4. THE PROPOSED INTEGRATED FRAMEWORK

In figure 1, the three major phases and six logical activities provide the skeleton, methodical and structured way of software development. The four major phases of RUP are now reduced to three and nine major logical activities of RUP are now reduced to six to make more efficient the process and to provide simplicity. All of the six logical activities of RUP can be used in each phase of this proposed hybrid model. Business investigation and Design is the main player in Domain Analysis and Design phase. The Implementation and Testing are the main logical activities of ‘Production and Validation’ phase, where as the Deployment and Configuration are the logical activities in Evolution phase. This proposed framework not only provides the skeleton and structured approach through its phases and logical activities but also supports engineering points of view with some of its very useful activities along with XP practices whose detail is given below in logical activities for each phase.

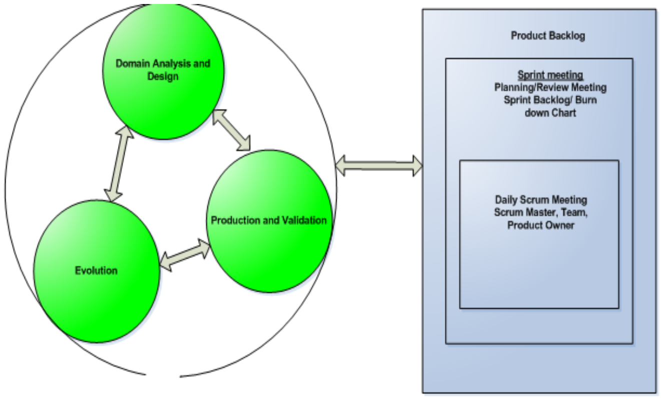
Domain Analysis and Design
Production and Validation
Evolution
Product Backlog
Sprint meeting
Planning/Review Meeting
Sprint Backlog/ Burn down Chart
Daily Scrum Meeting
Scrum Master, Team, Product Owner

Figure 1 The Proposed Integrated Framework

Formal procedures (Daily Scrum Meeting, Sprint Planning Meeting and Sprint Review Meeting) and job responsibilities (SM, Developers, Product owner) and objects (Product Backlog, Sprint Backlog, Burn down Chart) of Scrum can be fixed into the proposed model without any difficulty. The daily scrum meeting, sprint planning meeting and sprint review meeting can be conducted iteratively in each phase of proposed model. Their detail is given in development cycle.

XP, on the other hand, provides best engineering practices (Simple design, User stories, Metaphor, Pair programming, Collective ownership, Coding standards, Integration, Test driven development, On-site customer and 40-hours work) can also be fixed into the proposed model’s phases in different logical activities without any difficulty.

Another important factor of having documentation is that, as XP and Scrum welcomes change during the course of software development with known architecture and more often facilitates in case of an existing system development. That’s why; this change management doesn’t cross the boundary and disturb the design and architecture of existing system. But, when we deal with new system under the guidelines of XP and Scrum, we must have some mechanism that guides us how to manage the change request frequently. This change request mechanism leads towards the modeling at minimum level not only used to manage changes frequently but also helps to identify some major risks associated with application and to handle non-functional requirements effectively.

5. CONCLUSION

Document-driven approaches for software development provides a very structured and formalized process for software development through its deep planning, analysis and design, codified process and heavy documentation. Agile process models follows an iterative and incremental approach, heavily focus on constant customer collaboration delivering early release through small iterations, provides low bug rates, and frequent adaptation of changing business requirements.

RUP is an object oriented process model that takes an incremental and iterative approach and is based on sound software engineering principles. It describes well defined disciplines that acts as a skeleton for all kinds of projects especially for large scale but doesn’t give best engineering practices in order to achieve simplicity, reliability and quick adaptation to changing requirements. Scrum provides best mechanism to manage and track the application development but lacks of structured approach and best engineering practices. XP has best engineering practices but lack of structured and formalized nature as well as lack of best managerial aspects.

Our core objective of this research is to get high quality software that should be according to schedule, meet customer expectations at higher rates and increase team productivity. For this, there is a novel Hybrid framework that integrates the strengths of XP, RUP and Scrum methodologies while suppressing their weaknesses. Scrum will provide best managerial practices through its ceremonies, roles and artifacts throughout SDLC. RUP is used not only to provide skeleton, structured and formalized way throughout SDLC but also strengthen XP practices through its philosophy. This novel framework will also highlight some of very useful practices from RUP, but their applicability depends on the nature and type of project. XP is too much famous for its engineering practices like pair programming, user stories, test driven development as well as productivity enhancement activity i.e. 40 hours work.

REFERENCES

[1] Juyun cho. A Hybrid Software Development Method For Large Scale Projects: Rational Unified Process with Scrum. Issues in Information Systems. **10(2):**

341348, 2009.

[2] Ambler, S. A manager's introduction to the Rational Unified Process (RUP). Retrieved Feb, 2009, http://www.ambysoft.com/downloads/managersIntroToRUP.pdf.

[3] Sillitti, A., Ceschi, M., Russo, B., et al. 'Managing uncertainty in requirements: a survey in documentation-driven and agile companies'. Proc. Software Metrics 2005, 11th IEEE Int. Symp., 17-27, 2005.

[4] Dennis, A., Wixom, B. H., & Tegarden, D. Systems analysis and design with UML version 2.0. Hoboken, NJ: Wiley, 2005.

[5] Kruchten, P. The rational unified process: An introduction. Reading, MA: Addison-Wesley Longman Inc, 2004.

[6] Paetsch, F., A. Eberlein, and F. Maurer, "Requirements Engineering and Agile Software Development," Proceedings of the 12th IEEE international Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 308-313, 2003.

[7] Bard Appleton, Steve Konieczka, "Agile" Change Management – from first principles to best practices. CM Crossroads, 2004.

[8] Ionel, Naftanaila, Critical Analysis of the Scrum Project Management Methodology. 435-441, 2008.

[9] Mike Arvela, Matias Muhonen "Agile Requirements Development". 2006.

[10] Boehm, B. & Tuner, R. Management challenges to implementing agile processes in traditional development organizations, Software, IEEE, **22(5):** 20-29, 2005.

[11] Boehm, B & Papaccio, P. Understanding and controlling software costs. IEEE Transactions on Software Engineering. **14(10):** 1462-1477, 1988.

[12] Schach, S.R. An Introduction to Object-Oriented Systems Analysis and Design with UML and the Unified Process. Boston: McGraw-Hill, 2004.

[13] Boehm, B. & Tuner, R. Using risk to balance agile and plan-driven methods. IEEE Computer, **36(6):** 57-66, 2003.

[14] Bowers, J., May J., Melander, E., Baarman, M. & Ayoob A. Tailoring XP for large systems mission critical software development, Proceedings of the second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods – XP/Agile Universe 2002, 100-111, 2002.

[15] Hanakawa, N. & Okura, K. A project management support tool using communication for agile software development, Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC'04), 316-323, 2004.

[16] Lippert, M., Becker-Pechau, P., Breitling, H., Koch, J., Kornstadt, A., Roock, S., Schmolitzky, A., Wolf, H., & Zullinghoven, H. Developing complex projects using XP with extensions, Computer, **36(6)**: 67-73, 2003.

[17] Li Jiang, Armin Eberlin. Towards a framework for understanding the relationship between classical software engineering and agile methodologies. ACM, 9-14, 2008.

[18] Schwaber, K. Agile project management with Scrum. Redmond, WA: Microsoft Press, 2004.

[19] Paul Prior, Frank Keenan. Requirements Management in a Distributed Agile Environment. WEC **(2):** 204-207, 2005.

[20] John Hutchinson, Gerald Kotonya, Ian Sommerville, Steve Hall: A Service Model for Component-Based Development. EUROMICRO, 162-169, 2004.

[21] Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2005). Object-oriented analysis & design with unified process. Boston: Thomson Course-Technology.

[22] Cao, L., Mohan, K., Xu, P. & Ramesh B. How extreme does programming have to be? Adapting XP practices to large-scale projects, Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS'04), May 15-21, Hawaii, 2004.