I.J. Information Technology and Computer Science, **2012, 8, 43-50**
Published Online July 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2012.08.05

Empirical Estimation of Hybrid Model: A Controlled Case Study

Sadaf Un Nisa
Department of Computer Science, Virtual University of Pakistan, Defence Road, Off Raiwind Road Lahore, Pakistan
Email: ms080400004@vu.edu.pk

M. Rizwan Jameel Qureshi
Department of Information Technology, Faculty of Computing and Information Technology, King Abdul-Aziz University, Jeddah, Saudi Arabia
Email: rmuhammd@kau.edu.sa

***Abstract*—**Scrum and Extreme Programming (XP) are frequently used models among all agile models whereas Rational Unified Process (RUP) is one of the widely used conventional plan driven software development models. The agile and plan driven approaches both have their own strengths and weaknesses. Although RUP model has certain drawbacks, such as tendency to be over budgeted, slow in adaptation to rapidly changing requirements and reputation of being impractical for small and fast paced projects. XP model has certain drawbacks such as weak documentation and poor performance for medium and large development projects. XP has a concrete set of engineering practices that emphasizes on team work where managers, customers and developers are all equal partners in collaborative teams. Scrum is more concerned with the project management. It has seven practices namely Scrum Master, Scrum teams, Product Backlog, Sprint, Sprint Planning Meeting, Daily Scrum Meeting and Sprint Review. Keeping above mentioned context in view, this paper intends to propose a hybrid model naming SPRUP model by combining strengths of Scrum, XP and RUP by eliminating their weaknesses to produce high quality software. The proposed SPRUP model is validated through a controlled case study.

***Index Terms*—**Agile Models, Scrum, XP, RUP, Quality

I. Introduction

The goal of process models is to establish a relationship between quality of the product and quality of process [1][2].Scrum and Extreme Programming (XP) are frequently used agile models where as Rational Unified Process (RUP) is a conventional plan driven software development model. Each of the agile and plan driven approaches have their own strengths and weaknesses. A large number of software development companies are trying to integrate agile models with traditional plan driven models to enrich the benefits of both approaches to suppress their limitations from last many years [1]. Using agile methods projects are delivered on time. Kruchten [3] describes about RUP that it can be adopted as a whole or as a part because RUP gives adoption relaxation. In many cases, however a thorough configuration i.e. modification of RUP is suggested before its implementation. The configuration yields a development case that lists all deviations need to be made with respect to RUP model. RUP is not generally considered agile in nature. The method was originally developed to cater for the entire software production process, and therefore contains extensive guidelines for process phases that are next to irrelevant in an environment that is likely to call for an agile approach. [4] is written to examine the potentials of RUP for its agile adaptation.

The idea, behind the proposal of SPRUP model, is that the XP provides a very effective project engineering abilities, Scrum provides effective project management framework (where both of they are agile development methods) and RUP is most document and plan driven model. The integration of Scrum, XP and RUP will enrich their benefits to focus on satisfying business and customer needs. The resultant outcome will be an efficient model i.e., SPRUP that contains management, engineering and productive capabilities. Scrum is a framework, not a well defined process or system development life cycle (SDLC), and it leaves very much more on the development team rather than providing complete and detailed descriptions or proper plan for the project. The main advantage of doing so is that the team knows best how to solve the problem related to the project. XP have engineering practices enabling a development team to deliver more reliable software to the customer on time through the continuous feedback. This reduces the rework costs and efforts substantially and improves confidence among team members. A good feature of RUP model is that RUP works well for medium and large scale projects. Although RUP model has certain drawbacks, such as tendency to be over budgeted, slow in adaptation to rapidly changing requirements and reputation of being impractical for small and fast paced projects. The strengths of Scrum and XP are intended to add into SPRUP model. RUP lacks in adaptation to rapidly changing requirements due to well established requirement documentation. This feature is not lacking in Scrum and XP models as they respond well to creeping requirements. The main strength of RUP model is that it focuses on satisfying business and customer needs by delivering quality software and gives comprehensive planning for the system.

XP model has certain drawbacks such as weak documentation and poor performance for medium and large development projects. In SPRUP model this feature is tried to eliminate by adding RUP phases. XP has a concrete set of engineering practices that emphasizes on team work where managers, customers and developers are all equal partners in collaborative team. Scrum is more concerned with the project management. It has seven practices namely Scrum Master, Scrum teams, Product Backlog, Sprint, Sprint Planning Meeting, Daily Scrum Meeting and Sprint Review. Keeping above mentioned context in view, this research is intended to propose a model by combining strengths of Scrum, XP and RUP as well as narrower the weaknesses to produce quality software that adapt changing requirement quickly.

Scrum and XP focus on iterative and incremental development, customer collaboration, and frequent delivery through a light and fast development life cycle. The main motivation and beauty of SPRUP model is that it has a flavor of combining above mentioned strengths of Scrum, XP and RUP as well as narrower their weaknesses to produce quality software that adapt changing requirement quickly.

The paper is further organized as: section 2 covers related work. Section 3 defines the research. Section 4 describes the research design. Section 5 illustrates the proposed solution. Section 6 provides the evaluation of the proposed solution using a controlled case study. Conclusion and future work are given in the final section.

II. Related Work

Usually processes do have people satisfaction particular the aspect of roles to work jointly to bring into being artifacts, which could be the source code, architecture diagrams, and requirements documents [5]. In addition, these artifacts more often than not demarcate milestones in the project, which each process acknowledges in different ways [6]. Each process overview generally follows the same outline, due to which it becomes easier to compare the different processes along common criteria [7]. The pattern contains the detailed description of the process as well as unique characteristics from the main texts explaining the process.

- Roles – What specific positions are called for in the process and how are they filled?
- Artifacts – What kinds of documents and other artifacts are produced, how often they are produced, and how critical they are to the process [8]?
- Tools Support – How many different kinds of tools are available for using the process and what is the cost?

Rational Unified Process (RUP) provides a disciplined approach to assign tasks and responsibilities within a development organization. Its goal is to ensure the production of high quality software meeting the needs of its end users within a predictable schedule and budget [3]. Unlike the other processes discussed in this paper, RUP is a process product, which means that it is developed, maintained, and sold by Rational Software (now owned by IBM). An organization cannot fully use RUP without accompanying product, although it can use the unified process, which is the precursor and basis of the Rational Unified Process, as described in [9]. Additionally, you do not have to use rational products for every aspect of the Rational Unified Process.

Agile models are getting popularity from last many years. The general principle behind XP is that software is nothing without the code [10]. XP includes detail plans and designs that is one of the reasons why XP has drawn some criticism from opponents. Scrum is not so much a development process as a management process [11]. Scrum can be wrapped around an existing development process, and it has been used quite successfully in combination with XP [12]. By itself, Scrum cannot produce software; it must be combined with another development process [13]. It can simply organize how the software is developed.

III. Research Problem

A number of agile models are experienced by combining them with conventional software development models to maximize the strengths of both agile and conventional models while trying to suppressing the weakness of each approach [14][15]. However there are many combinations that are still inexperienced. Integrating Scrum, XP and RUP models is a good combination to accommodate the features of both conventional and agile models. Advantage of RUP model is that it focuses on satisfying business and customer needs by delivering quality software and gives comprehensive planning for the system. Major drawback of RUP model is that it fails to provide clear implementation guidelines and leaves the tailoring to the user entirely. XP and Scrum are self managed techniques through iterative planning. XP model has certain drawbacks such as weak documentation and poor performance for medium and large development projects. Scrum is more concerned with the project management [16][17]. Keeping above mentioned context in view, the research problem becomes:

How to propose a new novel hybrid model by combining strengths of Scrum, XP and RUP as well as *narrower the weaknesses to produce quality software to adapt changing requirements quickly?*

IV. Research Design

Qualitative methods are categorized as observation, case study, interview, document analysis, open-ended questionnaires and focus groups. Qualitative methodology includes action research, ethnography and discourse analysis [18]. Cormack [19] explains that qualitative research techniques include participant observation, in-depth interviews, focus groups, oral histories and conversational analysis. The author Cormack says that in-depth interview is very useful in obtaining the experiences of the respondents which they describe in their own words. In-depth interviews may be semi-structured or unstructured. Focus groups are useful when the researchers need data pertinent to the cultural values of a group. Participant observation is very much beneficial when the researcher would like to obtain data relating to behaviors that occur naturally in their usual contexts [19]. Cooper and Schindler [20] defined the qualitative research as "An array of interpretive techniques which seeks to describe, decode, translate and otherwise come to terms with the meaning, not the frequency of certain more or less naturally occurring phenomenon in the social world" [20].

The basic purpose of conducting case study was to build a system which is based on the implementation directions of the proposed SPRUP model. For this purpose a team consisting of 5 members was selected. The duration of case study was stipulated to five weeks. During this period, the team generated four SPRUP releases, comprehensively working under the guidelines proposed in SPRUP model. At the time of selection of team it was kept in mind that the team was balanced from all aspects such as designing, coding, testing etc. The researchers were in positions to implement and design the environment of the case study during this research. Therefore, this case study was controlled in nature. The role of Scrum Master for conducting the daily scrum meetings and monitoring whether the development is moving exactly according to the guideline of proposed SPRUP model.

The team worked under the supervision of Project Manager and SPRUP master. The conclusion showed that SPRUP model enhanced the performance, quality and productivity of the delivered product. For this purpose, a team of 5 members was selected. The duration of case study was fixed five weeks. The project assigned to development team was Hostel Management System for a famous educational institute. The team worked under the guidelines and directions mentioned in the proposed hybrid SPRUP model. It took four iterations to complete the project. Before starting development, a training program was conducted to educate the team about the proposed model.

The basic purpose of conducting case study was to build a system which is based on the implementation directions of the proposed SPRUP model. The duration of case study was stipulated to five weeks. During this period, the team generated four SPRUP releases, comprehensively working under the guidelines proposed in SPRUP model. At the time of selection of team it was kept in mind that the team was balanced from all aspects such as designing, coding, testing etc. In this case study researchers were in position to implement and design the environment of the case study therefore this case study was controlled in nature. The authors of this research were performing the roles of 'Scrum Master', 'Product Owner' and 'Project Manager'.

V. The Proposed Hybrid Model

SPRUP is intended to merge the project management strength of Scrum, production practices of XP and customer satisfaction of RUP to produce high quality software that adapts to changing requirement quickly and meets the business goals. The aim, behind constructing SPRUP model, was to have a model that could manufacture products with high quality and low defect rate. Inception, Elaboration, Construction and Transition phases of RUP, combining features of product backlog and sprint backlog out of Scrum and strong agile practices of XP are adopted into SPRUP model. During SPRUP sprint cycle, anticipation, crafting, execution and assessment activities are added to perform the work done in planning, designing, coding and testing practices of XP model. SPRUP model includes SDLC activities of XP model along with process outline namely sprint out of Scrum. Complete implementation guideline regarding software development is provided in suggested SPRUP model using the phases of RUP. The main phases and activities, of SPRUP model, are shown in figure 1.

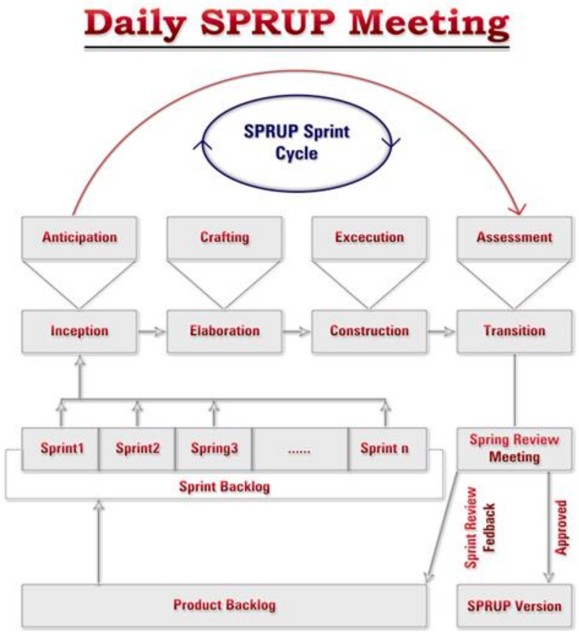

Fig 1: The Proposed Hybrid Model

SPRUP sprint cycle moves the software product towards its completion where product meets the desired quality measurements. Quality metrics includes customer satisfaction at its highest priority. SPRUP model contains the combined features of RUP, Scrum and XP. Where phases of RUP model as well as sprinting of Scrum along with rich features of XP provides the basis for the SPRUP model. The duration of sprint cycle in SPRUP should be at-least of 1 week long or two weeks long at its most. To achieve the feature of well-timed delivery of working set of product to the customer, involvement of product owner is in SPRUP sprint cycle is kept high. In SPRUP, the duration of sprint cycle in SPRUP is at-least of 1 week long or two weeks long at its most. The foremost thing in SPRUP cycle is done by the product owner. Product owner gives broad-spectrum plan for the project description about the road map of the development process to the SPRUP product development team.

When the general sprint goals of development plan are done by the product owner and are also agreed by the SPRUP master then both of them, with the help of development team create the product backlog. Product backlog provides the basic functionality of the product. Once the product backlog is finalized by product owner then it is the job of development team to select the sprint backlog and decide prioritized list of tasks to be performed during a single sprint cycle. Prioritization done by development team is within a single sprint. Product owner does the prioritization of the items within the product backlog. The process of prioritizing among the items in sprint or product backlog also involves discussion with customer and SPRUP master.

It is important to involve the customer in the prioritization discussion because some of the times customers need pieces of product (working set of software) on priority. Therefore prioritization is done through combined effort. In the end the process is finalized by product owner. Post prioritization (prioritization among product backlog first and sprint backlog afterward) step in SPRUP model is estimation of time expected to be spent on tasks of a sprint i.e. time for sprint cycle to be started in hrs. After defining and estimating the tasks for sprint in sequence RUP phases are applied. Developers start their work by following the activities and features of 1: inception, 2: elaboration, 3: construction and 4: transition in sequence. SPRUP team completes defined task following these phases within time in each sprint cycle.

In inception phase requirements are collected from the customer in the form of user stories. In the second phase of elaboration functional requirements and non-functional requirements are collectively defined. Actual code written activity against the tasks of sprint is done in construction phase of SPRUP cycle. Out of four phases of SPRUP cycle lastly transition is done. Transition phase includes tasks like daily based testing and validating activity of system. Such activities are done in SDLC with the purpose to minimize the defects relating to the software product. Such activities in SPRUP cycle are highly beneficial as a part of daily meetings because earlier detection and fixation of defects is highly appreciated in software development world. Product backlog is organized by the product owner at the beginning of SPRUP model. The Product owner is responsible to prepare product backlog. All of functional, non functional, statistical and economical requirements of the system under development are gathered by the customer. After deciding down the general goals of the system product backlog is created by the product backlog. Continuous updates are supplementary for product backlog in accordance with Review Feedback of sprint or Sprint Review Feedback (SRF) collected during sprint review meeting. Product backlog is maintained by the product owner throughout all of the SPRUP sprint cycles.

Maintaining the sprint backlog is the job of SPRUP Master. Duration of each sprint is kept between one or two weeks. Idea behind short sprint cycle is to regularly collect the creeping requirements through sprint review feedback at the end of each sprint. Number of sprints required to complete the project are decided by the SPRUP master, project developers (development team) and product owner. At the completion of one sprint cycle, sprint version is released. Product backlog items are moved to the sprint backlog for implementation. SPRUP Master with the help of product owner and SPRUP team decides down the sprint backlog. In product backlog items can be changed through SRF in contrast items in sprint backlog before the release of SPRUP version and completion of SPRUP cyclic activities cannot be changed.

Role of SPRUP master is to identify the development progress regularly. Purpose of doing so is to keep the development towards the achievement of goals defined in inception phase of model. SPRUP master is responsible to inquire the individual development progress from team members. This interaction of development team and SPRUP master is done in daily SPRUP meeting. Another job of SPRUP master is to remove the problems or troubles faced by the development team. SPRUP master also motivates the rest of the members by telling them that development process is in its accurate track.

SPRUP master is also responsible for conducting and organizing sprint planning meeting. It is done to plan about the sprint. The participation in meeting vary from sprint to sprint mainly SPRUP master likes to invite product owner along with development team, but in some of the sprints customer involvement becomes necessary. Customer involvement is usually dependent on the last SRF given by the customer, so customer is also expected participant in the sprint planning meeting. For small sprints this meeting can be conducted or concatenated with the anticipation activity of inception phase. But this decision of concatenation is to be taken by the SPRUP master.

Four phases having mainly four activities are followed in sequence during daily SPRUP meetings. These daily meetings are conducted in daily basis. Tasks are assigned to the development team on daily basis and evaluation of previously assigned tasks is also done on daily basis. Development team along with SPRUP master takes part in daily meetings. Inception, elaboration, construction and transition phases are executed in sequence of meetings. Time duration of meeting is from 15 to 30 minutes. The main advantages of daily meetings are:

- It provides a platform for development team to discuss their development related issues.
- It helps in measuring the work progress of team.
- Barriers, related to progress of work, are removed quickly.

At the end of each sprint, sprint review meeting is conducted in which results of the new deliverable are provided to the customer, management and product owner by SPRUP master and team. After completing 1st three phases of SPRUP cycle a sprint review meeting is conducted as a part of fourth phase of SPRUP cycle i.e. transition phase. This review meeting is held to conclude the sprint. Sprint Review Feedback (SRF) is collected as an outcome of the meeting.

Another purpose of review meeting is to evaluate any deficiencies or discrepancies in the sprint backlog. If there is any, then it is given to the product backlog in the form of sprint review feedback. If all the activities of phases with their tasks are carried out without any deficiency and SPRUP sprint cycle finalizes its iteration successfully, SPRUP version is released. This version approaches an increment in development. Release of SPRUP version is highly dependent on degree of customer satisfaction and approval by customer. The approval of the SPRUP version depends upon the extent of the customer satisfaction. When transition phase ends successfully then SPRUP version is released.

On successful release of SPRUP version, the SPRUP sprint cycle restarts from the inception phase for next sprint. Participation of product owner was appreciating because this feature helps in maximizing the customer satisfaction in product to be developed. Prioritization of next sprint is also discussed at the end of this meeting. Working set of the product released to the customer is called SPRUP version which is released after approval in sprint review meeting. After the successful completion of all sprints the whole product is commenced with its all features and functionalities. Work done during the sprint is closely observed in Sprint review meeting. Any change or new suggestions are raised during sprint review meeting. During this meeting changing suggestions is part of meeting out come. Suggestions and changes are considered /collected as sprint review feedback. They are in the form of new product attributes or in the form of complete backlog, and will be updated to the product backlog. Change in requirements by the customer is also possible during sprint review meeting while examining the evaluation process. The new product attributes are considered the part of SRF. In SPRUP model SRF becomes the part of product backlog out of which suggestions and requirement change is applied on items in product backlog. SRF helps in improving the outcome of next sprint.

VI. Evaluation and Discussion

A controlled case study was conducted to validate the SPRUP model. The researchers were in positions to design the implementation environment before starting the case study. Four releases called ‘SPRUP versions’ were produced during this case study. The first version was of two weeks duration. Rest of three versions was of one week duration. The case study data is collected from four sprint versions. The collected data is represented in the tabular form in the Table 1. The first column represents the attributes to be measured for each version. The last column contains the cumulative/average data from all the four releases. Each row represents a specific attribute to be measured in each SPRUP version. All the columns represent sum/ average data from versions while each line of table represents data of all releases of a particular parameter of the case study. Table 1 shows the details of the data about the case study.

The first version (Column one in the Table 1) was completed in two weeks, whereas each of remaining three versions took one week duration. The term ‘SPRUP version’ is used in SPRUP model that is about the working set of product developed for actual customer use. The number of modules in row two of Table 1 (built during development process of sprint) is represented in each SPRUP version. Total tasks defined within modules are represented in row 3. Each version shows number of tasks defined in their respective columns. Total work effort is in row 4. Total work effort of the project is remained constant throughout all releases. Work effort and customer involvement are calculated using following formulas

Work Effort = (working days in a week/week) * (duration in weeks/release) * (team size)* (working hrs/ day) (1)

Customer Involvement = Time spent by customer (hr)/ Task allocated actual hours * 100 (2)

Work hr per day were 8, where as 5 working days were considered per week. The total actual hours dedicated by the developer to tasks are given in row five. It was 350 at first the release and in 2nd and 3rd version/release it was 160 to 170 hr respectively. Percentage of task effort is calculated in row 6 of Table 1. Reduction of task effort was from 87% during 1st version to 80-85% in 2nd and 3rd versions respectively and was 55 % in 4th version. Average of all four versions was 76%. It showed that short development cycles cause an increase in over head.

The number of interfaces for each version is given in row 7. Total interfaces built during the development were 49. The Line of Code in total against all of four versions is 38005. During whole development cycle totality of number of classes built was 69, with Line of Code 4240. Total number of 29 test classes built for testing purpose having 19323 Line of Code (LOC). The amount of total lines of code the team produced in a release is represented in rows 11 and 12. Pre-release defects are mentioned in row 15. Total defect rate was lower comparatively i.e. 0.489 defects/KLOC were calculated. In addition, 16 improvement suggestions are gathered during SRF (Sprint Review feedback). Most of the suggestions are raised from the first two releases.

In this controlled case study, the customer was ideally available in the same environment with the development team and the actual customer involvement (row 20) was only 25% on average.

Table 1 Data Gathered During the Case Study

		Releases ⟶				
ID	**Parameters ↓**	**1**	**2**	**3**	**4**	**Average**
1	Time (weeks)	2	1	1	1	5
2	Items per Sprint backlog	9	4	4	3	20
3	Total Tasks defined	40	22	14	5	81
4	Total work effort (hr)	400	200	200	200	200
5	Task allocated actual hours	350	160	170	110	790
6	Task allocated actual (%)	87	80	85	55	76%
7	Interfaces	29	7	8	5	49
8	Classes	55	6	5	3	69
9	Test Classes	12	4	7	6	29
10	Test Classes LOC	10518	2380	2246	4179	19323
11	Total LOC	19963	6581	4010	7451	38005
12	Total KLOC	19.963	6.581	4.010	7.451	3.8005
13	Team Productivity (LOC/H)	57.04	25.05	25.68	66.36	47.37
14	Number of Integration	51	19	29	22	121
15	Pre-Release Defects	5	2	2	1	10
16	Post release defects	7	5	3	1	16
17	Post release defects /KLOC	0.35	0.456	0.75	0.40	0.489, 1.956
18	Sprint Review Feedback	7	5	3	1	16
19	Pair programming %	80	80	80	80	80%
20	Customer involvement %	35	25	20	20	25%
21	User Stories/ Requirements	21	9	3	2	35
22	Unit Tests	113	45	15	25	198
23	Customer Satisfaction%	80	80	90	90	85%

Customer satisfaction is measured about how products and services provided by the company fulfill the customer’s expectation. The user stories described in each release are given in row 21. Row 22 deals with the testing activity. In this case study, the customer satisfaction is measured in terms of satisfaction over number of modules of the product. The row 23 represents customer satisfaction in percentage form. Level of customer satisfaction remained 80% during the 1st and 2nd versions. The satisfaction level rose to 90% for the 3rd and 4th versions.

VII. Conclusion

The motivation for the proposal of SPRUP software development model is to combine the rich features of XP, Scrum and RUP. XP provides a very effective project engineering abilities and Scrum provides effective project management framework and both of XP and Scrum are agile development methods. These models are enriched with the RUP model that focuses on satisfying business and customer needs and the outcome is an achievement in the form of high customer satisfaction. The main enriching feature of SPRUP is that strengths of Scrum, XP and RUP as well as narrower their weaknesses to produce quality software that adapt changing requirement quickly.

Scrum is a framework, not a well defined process or SDLC. Therefore, Scrum does not provide proper plan for the project and most of the project management activities are left on the developers. Advantage of doing so is that the team knows better about their working capabilities, developmental environment and problems specifically relating to a project. Furthermore Scrums' management inheritance towards development team is being combined with XP where XP have engineering practices. This combination of Scrum and XP helps effectively in reducing rework cost and effort which is highly discouraging in software world. A good feature of RUP model is that RUP works well for medium and large scale projects. Although RUP model has certain drawbacks, such as tendency to be over budgeted, slow in adaptation to rapidly changing requirements and reputation of being impractical for small and fast paced projects. The strengths of Scrum and XP were intended to add into the proposed SPRUP model because both are self managed processes through iterative planning. RUP is slow in adaptation to rapidly changing requirements due to well established requirement documentation. This feature is not lacking in Scrum and XP models as they respond well to creeping requirements. The main advantage of RUP model is that it focuses on satisfying business and customer needs by delivering quality software and gives comprehensive planning for the system.

XP model has certain drawbacks such as weak documentation and poor performance for medium and large development projects. These limitations of XP model are eliminated by introducing the phases of RUP model into the proposed model. XP has a concrete set of engineering practices that emphasizes on team work where managers, customers and developers are all equal partners in collaborative team. Scrum is more concerned with the project management. It has seven practices namely Scrum Master, Scrum teams, Product Backlog, Sprint, Sprint Planning Meeting, Daily Scrum Meeting and Sprint Review.

Keeping above mentioned context in view, this paper proposed a model by combining strengths of Scrum, XP and RUP as well as narrower their weaknesses to produce quality software. The proposed SPRUP model was validated through a controlled case study. SPRUP is validated on a small scale project. It is further needed to be applied on a large scale project. It can also be enhanced to out sourcing development surroundings. More combinations of RUP and agile models are needed to be experienced and it can be a good future work of this research.

References

[1] Szalvay V. An introduction to agile software development, Retrieved June 2012 from http//www.danube.com/docs/Intro_to_Agile.pdf, 2004

[2] Larman L. Agile and Iterative Development: a manager's guide Iterative and Evolutionary. USA: Addison-Wesley Professional, Pearson Education Inc., 2004

[3] Kruchten P. The Rational Unified Process: An Introduction The Rational Unified Process. USA: Addison-Wesley Professional, Pearson Education Inc., 1999

[4] Schwaber K, Beedle M. Agile Software Development with Scrum: Advanced Development Methods. USA: Prentice Hall, Upper Saddle River, 2002

[5] Alfonso M I, Botia A. An Iterative and Agile Process Model for Teaching Software Engineering. In: Proceedings of the 18th Software Engineering Education and Training Conference, Ottawa, Canada, 2005.9-16

[6] Beck K, Andres C. Extreme Programming Explained: Embrace Change. What is XP?. USA: Addison-Wesley Professional, Pearson Education Inc., 2004

[7] Dagnino A. An Evolutionary Lifecycle Model with Agile Practices for Software Development at ABB. In: Proceedings of the 8th IEEE International Conference on Engg. Complex Comp. Sys. Maryland, USA, 2002.215-223

[8] Hossain E, Baber M A, Paik H. Using Scrum in Global Software Development: A Systematic Literature Review. In: Proceedings of the 4th IEEE International Conference on Global Software Engineering, Limerick, Ireland, 2009.175-184.

[9] Jacobson, I, Christerson M, Jonsson P, Overgaard G. Object Oriented Software Engineering: A Use-Case Driven Approach. USA: Addison-Wesley,1992

[10] Schwaber K. Scrum Development Process. USA: Middlesex Turnpike,1996

[11] Judy K H, Krumins-Beens I. Great Scrums Need Great Product Owners: Unbounded Collaboration and Collective Product Ownership. In: Proceedings of the 41st Hawaii International Conference on System Sciences, Waikoloa HI, USA, 2008.462-462

[12] Qumer A, Henderson-Sellers B. Comparative Evaluation of XP and Scrum using the 4D Analytical Tool (4-DAT). In: Proceedings of the European and Mediterranean Conference on Information System, Costa Blanca, Alicante, Spain, 2006.1-8

[13] Schwaber K. Agile Project Management with Scrum. Backdrop: The Science of Scrum. USA: Microsoft Press,2004

[14] Cockburn A. Agile Software Development: Evolution of Agile Methodologies. India: Pearson,2002

[15] Cockburn A, Highsmith J. Agile Software Development: The People Factor. IEEE Computer, 2002, 34 (11): 2002131-133

[16] Marchesi M, Mannaro K, Uras S, Locci M. Distributed Scrum in Research Project Management. In: Proceedings of the 8th International Conference on Agile processes in software engineering and extreme programming, Como, Italy,2007.240–244

[17] Kamlesh V, Ahmad S. Evaluating Evolutionary Prototyping for Customizable Generic Products in Industry. M.S. Thesis, School of Engg. Blekinge Inst. Tech. (Ronneby, Sweden), 2008

[18] Truscott D M, Swars S, Smith S, Thornton-Reid F, Zhao Y, Dooley C, Williams B, Hart L, Mathews M. A cross-disciplinary examination of the prevalence of mixed methods in educational research. International Journal of Social Research Methodology. 2010, 13 (4):317–328

[19] Cormack D. The Research Process in Nursing: Common terms and concepts in research. USA: Blackwell Publishing Inc. 2000

[20] Cooper D R, Schindler P S. Business Research Methods. India: Tata McGraw-Hill Publishing.2005

Sadaf Un Nisa: received her MS degree from Virtual University, Pakistan in 2011. Her main interest is software process improvement under software engineering domain.

Dr. M. Rizwan Jameel Qureshi: received his Ph.D. degree from National College of Business Administration & Economics, Pakistan in 2009. He is an author of thirty six publications at National and International forums. His major interest areas are component-based software development, agile software development, cloud computing and agent oriented software engineering. He had over thirteen years of professional experience at top universities of national and international levels. He is currently working as an assistant professor at IT Dept., King Abdul Aziz University Jeddah, Saudi Arabia.