

Improved Scrum Framework using Agile Features

Bakr Khaled Abdallah, Samir Fathy Ibrahim

M.Sc. Student, Computer Science, Neelain University, Sudan
bkabdallah88@hotmail.com

Basheer Abdalrahman

Lecturer, Information Systems, Neelain University, Sudan
rahmanbasho@hotmail.com

Abstract

This paper provides an enhanced Scrum framework that combines some practices of eXtreme Programming (XP) approach in Scrum framework to gain quality software on time. XP and Scrum are two agile software development methods. While Scrum is focused on project management, XP is focused on Software development; nevertheless, they both can be used to participate in the development of any software project independently or together. This paper presents the main concepts, features, phases, artifacts, and roles of Scrum as well as a brief introduction to XP and its practices. In this paper, the researchers present how to combine some XP practices into Scrum activities. The researchers exploit the features and best practices of the two methods to propose an improved Scrum framework that includes guidelines for achieving each Scrum activity or process. So, the improved Scrum framework is more applicable than many previous attempts in this domain. The improved Scrum framework has been validated by a group of 40 experts and specialists in agile software projects.

Keywords: Scrum; Agile Methods Software Development; XP Approach; Agile Best Practices; Software Project.

1- Introduction and Problem Definition

Recently, agile software development methods have gained popularity and are increasingly important to a significant number of software development organizations [2]. The use of agile methodologies enables software developers to produce higher-quality software in a shorter period [8]. Agile software development methods include XP, Scrum, Crystal, Feature Driven Development (FDD), Dynamic System Development Methodology (DSDM), and Adaptive Software Development (ASD) [4].

Scrum was introduced by Ken Schwaber and Jeff Sutherland in 1995. They also held an implementation workshop at OOPSLA '95 in Austin, Texas [25]. It is an agile software development framework that is widely used to achieve the agility, iterative, and incremental development in the software field. It is used for managing software projects in a changing environment [21]. Scrum is not a process or a technique for building products; rather, it is a framework within which we can employ various processes and techniques [13]. Scrum has the advantage of being very visible and its focus on functional software shows results to management that they can see and get excited about [24]. It increases revenues generated through the provided software. A well-functioning Scrum will deliver the highest business value features first and will avoid building features that will never be used by the customer [6]. Scrum has been adopted by large companies such as Yahoo!, Microsoft, Intel, and Nokia [2].

XP was developed at Chrysler by Kent Beck while working on a payroll project as a member of a 15-person team. Beck continued to refine and improve the XP methodology after the project was completed until it gained worldwide acceptance in 2000 and 2001 [8]. XP software development process focuses on iterative and rapid development [21]. XP approach stresses communication and coordination among the team members always; and requires cooperation between the customer,

management and development team to form the supportive business culture for the successful implementation of XP [1]. XP is designed for use in an environment of rapidly changing requirements.

While Scrum is focused on project management, XP is focused on software development. Each of them is used independently to produce software products. Any software project needs both management skills and software engineering practices. Therefore, focusing on either management skills or software engineering practices doesn't guarantee the success of the software projects. In other words, in case of using Scrum individually, the absence of good practices of software development will not lead to a quality software product even though there are good management skills. Similarly, in case of using XP individually, the absence of good management skills will not lead to a quality software product. Many research and surveys proved that Scrum and XP provide complementary practices and rules [3, 9, 11]. In addition, increasing number of surveys is supporting the anecdotal evidence that variations of combination of Scrum with XP are the most popular agile methods [26]. This finding is backed up by the results of surveys conducted by VersionOne that report on the status of organizations currently implementing or practicing agile methods.

Therefore, this paper proposes an improved Scrum framework that combines some XP practices into Scrum framework. The enhanced Scrum framework shows the cooperative relationship between Scrum and XP. The enhanced Scrum framework includes the Scrum activities and guidelines for achieving each activity. So, the improved Scrum framework is more applicable than many previous attempts in this domain. The improved Scrum framework may increase the success of software projects. The improved Scrum framework has been validated by a group of 40 experts and specialists in agile software projects.

2- Scrum Framework

Scrum is a project management framework that can manage and control software development. The Scrum name is derived from a strategy used in Rugby match where the ball is passed among team members to move the ball down the field. Scrum moves a project forward by improving communication and collaboration among team members and breaking the work into a series of sprints. In more detail, the Scrum framework consists of Scrum teams and their associated roles, events, artifacts, and rules. Each component within the framework serves a specific purpose and is essential to Scrum's success and usage [13]. Table (1) illustrates the components of the Scrum framework and their items that will be explained in more detail in the following subsections.

Figure (1) illustrates a graphical representation of the typical Scrum framework that shows its components. Depending on table (1) and figure (1), the researchers extract the main activities that are performed within the Scrum framework and relate its components. The Scrum activities are:

1. Preparing product backlog.
2. Sprint planning meeting and preparing sprint backlog.
3. Sprint.
4. Daily Scrum.
5. Sprint review and presenting an increment.
6. Sprint retrospective.

These activities represent a simple and cohesive framework that will be used by the researchers to explain his main contribution in sections (5) and (6). In the following subsections, a summary of each component of the typical scrum framework will be provided.

Table (1): Scrum Framework - The Components and their Items.

Scrum Component	Items or Description
Scrum Team	Scrum team includes of three roles: Product Owner, Development Team, and Scrum Master.
Scrum Events	Scrum events include of five events: Sprint, Sprint Planning Meeting, Daily Scrum, Sprint Review, and Sprint Retrospective.
Scrum Artifacts	Scrum artifacts include of three artifacts: Product Backlog, Sprint Backlog, and Increment.
The Rules of Scrum	The rules of Scrum bind together the events, roles, and artifacts, governing the relationships and interaction between them.

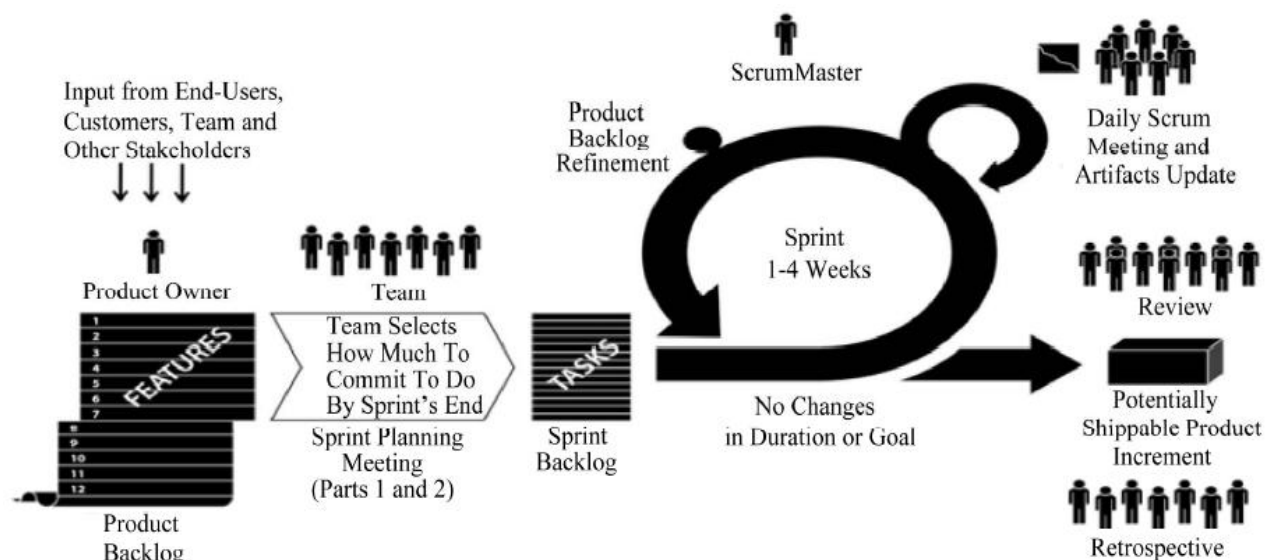


Figure (1): The Scrum Framework [23].

2-1 Scrum Team and Roles

Scrum Teams are self-organizing and cross-functional [13, 24]. Scrum team is designed to optimize flexibility, creativity, and productivity. Scrum team has three roles of people: product owner, Scrum master, and development team [25].

- Product owner is the person who is responsible for creating and prioritizing the Product Backlog, choosing what will be included in the next iteration/Sprint, and reviewing the system (with other stakeholders) at the end of the Sprint [15].
- Scrum master is responsible for ensuring that Scrum values, practices, and rules are understood and enforced. The Scrum master is responsible for facilitating Scrum events, conducting daily Scrum meeting, reviewing and evaluating sprint, removing obstacles, motivating the team, participating in product development, coaching development team.
- Development team is responsible for designing, building, and testing the desired product [14]. The development team is typically five to nine people in size; its members must have all the skills needed to produce quality software. Development team is self-organizing and cross-functional; thus the team members perform all design, development, and tests together. They have full authority to do whatever is necessary to achieve the sprint goal .

2-2 Scrum Events

Prescribed events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum [13]. Every event in Scrum has a time-box that implies a maximum duration. Scrum events consist of sprint, sprint planning meeting, daily Scrum, sprint review, and sprint retrospective.

- Sprint: The heart of Scrum is a Sprint, a time-box of one month or less during which a “Done”, useable, and potentially releasable product increment is created [13]. During the sprint, the team is supposed to have full authority over its actions and no external influence from the product owner, or anybody else, is allowed [2].
- Sprint planning meeting: During sprint planning, the product owner and development team agree on a sprint goal that defines what the upcoming sprint

is supposed to achieve [14]. Using the sprint goal, the development team reviews the product backlog to identify the highest priority items that will be included and accomplished in the upcoming sprint.

- **Daily Scrum:** Crucial to the sprint phase are Scrum meetings, which are held daily to determine the progress of the release and to respond to problems encountered along the way. Daily Scrum meeting is a 15-minute time-boxed event led by the Scrum master, who asks the same three questions to each team member every day to evaluate the development progress:
 1. What did you do yesterday?
 2. What will you do today?
 3. What's in your way?
- **Sprint review:** A sprint review is held at the end of the sprint to inspect the increment and adapt the product backlog if needed [13]. In the sprint review, the Scrum team and stakeholders collaborate about the work was done in the sprint and the next things that could be done. In the sprint review, the presentation of the increment is intended to elicit feedback and foster collaboration.
- **Sprint retrospective:** The Sprint Retrospective frequently occurs after the sprint review and before the next sprint planning [14]. It is useful for the Scrum team to inspect itself and identify the possible improvements to be considered during the upcoming sprint.

2-3 Scrum Artifacts

Scrum artifacts consist of product backlog, sprint backlog, and increment [13].

- **Product backlog:** The first activity of the product owner is to visualize the product and generate a refined and prioritized list of tasks which defines the product [25]. Then, she/he communicates it in the form of an ordered list known as the product backlog. The product backlog evolves as the product evolves. For ongoing

product development, the product backlog might also contain new features, changes to existing features, defects needing repair, technical improvements, and so on [14].

- **Sprint Backlog:** The Sprint Backlog is the set of Product Backlog items selected for the Sprint plus a plan for delivering the product Increment and realizing the Sprint Goal. The Sprint Backlog is a forecast by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality [13].
- **Increment:** The Increment is the total of all the product backlog items accomplished during the current sprint and all previous sprints. At the end of each sprint, the product owner will receive a deliverable, and will be able to see the incremental growing of the product [7].

3- XP Practices

XP is a software development methodology that does not rely on any particular tool, but rather is based on the common understanding of fundamental values and on a disciplined application of best practices [16, 22]. In addition, XP approach can be viewed as life cycle phases that include six phases. Table (2) illustrates the value, practices, and phases of XP approach. This paper focuses on XP practices.

Table (2): XP Approach – Values, Practices, and Phases.

XP Component	Items or Description
XP Value	XP is based on four values: • Simplicity • Communication • Feedback • Courage
XP Practices	XP values are implemented with twelve practices: • Planning Game • Small Releases • Metaphor • Simple Design • Testing • Refactoring • Pair Programming • Collective Code Ownership • Continuous Integration • 40-hour Week • On-site Customer • Coding Standard
Phases (XP Process)	The Phases of XP Approach: 1.Exploration 2.Planning 3.Iterations to Release 4.Productionizing 5.Maintenance 6.Death

XP approach is based on four main values: simplicity, communication, feedback, and courage. XP values are implemented with twelve core practices. There are mutual relationships among XP practices because each XP practice requires the other practices to be performed with an accepted level of quality.

- **Planning Game:** The planning game embodies the tension between what the customer wants and what the developers can deliver by allowing each of them to order the user stories by value and risk respectively [19]. The developers estimate the effort required to implement customers' stories and show the results to the customers to decide on the scope and timing of releases.
- **Small Releases:** The development is divided into a sequence of small iterations, each implementing new features separately testable by the customer [5]. These short iterations of 3-4 weeks accelerate the software delivery. At the end of a release, the customer reviews the software product to determine defects and adjust future requirements to be achieved in a new iteration. This process is

repeated a little number of times till an initial version of the software is put into production.

- **Metaphor:** The metaphor is used to guide all team members with a simple, shared story of how the overall system works [18]. It is an effective way of getting all members of the project team to visualize the project. It provides inspiration, vocabulary, and basic architecture of the system.
- **Simple Design:** In XP, developers use the simplest possible design that will satisfy the current needs. The right design of the software at any given time is the one that runs all the tests, has no duplicated logic, states every intention important to the programmers, and has the fewest possible classes and methods [10].
- **Testing:** Sometimes, this practice is called “test first” because tests in XP must be created prior to coding. All code must have automated unit tests and acceptance tests, and must pass all tests before it can be released [5]. The result is a program that becomes more and more confident over time.
- **Refactoring:** Refactoring is the process of restructuring the system without changing its behavior to remove duplication, improve communication, simplify, or add flexibility [18]. Refactoring throughout the development process saves time of development and increases quality.
- **Pair Programming:** Pair Programming means that two programmers work together to accomplish a development task using one shared computer. Pair programming provides an immediate peer review of code and its intended test case, and is one of the main reasons that XP induces quality into the code work product [4]. Moreover, it reduces the time required for task completion and it is useful in complex tasks and training.
- **Collective Code Ownership:** Once code and its associated tests are checked into the code base, the code can be altered by any team member. This collective code ownership provides each team member with the feeling of owning the whole

code base and prevents bottlenecks that might have been caused if the “owner” of a component was not available to make a necessary change [15].

- Continuous Integration: In XP, all changes are integrated and tested every few hours, or at least daily in the worst case [4]. All tests are run, and they have to be passed for accepting the changes in the code. XP team integrates and builds the software multiple times per day. Continuous integration helps the team to solve the development conflicts and participates in improving the quality of the development process.
- 40-Hour Weeks: In XP, the developers should not work more than 40-hour weeks. The people perform best and most creatively if they are rested, fresh, and healthy [22].
- On-Site Customer: A customer always works with the development team to answer questions, perform acceptance tests, and ensure that development is progressing as expected. This customer-driven software development led to a deep redefinition of the structure and features of the system [5]. It supports customer-developer communication [12].
- Coding Standards: This practice indicates that the developers must agree on a common set of rules enforcing how the system shall be coded. This makes the understanding easier and helps in producing a consistent code. Coding standards are almost unavoidable in XP, due to the continuous integration and collective ownership properties.

4- Previous Attempts of Combining Scrum and XP

There were many attempts to combine Scrum framework and XP approach. In the following, some of these attempts are briefly introduced:

- Jorge Edison Lascano dedicated a great part of his paper to provide an overview of software development phases, Scrum, and XP [9]. He presented the

characteristics shared by XP and Scrum. He considered that Scrum can be viewed as a wrapper to XP projects. XP fits inside the sprint phase in Scrum. In other words, XP practices: simple design, testing, refactoring, pair programming, collective ownership, continuous integration, and coding standards can be used inside the sprint phase. He didn't present how they can be used.

- Kane Mar and Ken Schwaber found that Scrum and XP provide complementary practices and rules [11]. They overlap at the planning game (XP) and Sprint planning (Scrum). Both encourage similar values, minimizing otherwise troublesome disconnects between management and developers. Combined, they provide a structure within which a customer can evolve a software product that best meets his or her needs and can implement quality functionality incrementally to take advantage of business opportunities. Following are several shared practices that facilitate this functionality: iterations, increments, emergence, self-organization, and collaboration. Iterations mean that all work is done iteratively, with the ability of the customer to direct the project through iterations. Increments mean that every iteration produces an increment of the customer's highest-priority functionality. If desired, the customer can direct the developers to turn these increments into live, operational functionality at any time. Emergence means that only the functionality that the customer has selected for the next iteration is considered and built. The customer doesn't pay for functionality that he or she might not select, and the developers don't have to code, debug, and maintain irrelevant code. Self-organization means that the customer says what he or she wants; development determines how much they can develop during an iteration and figures out the tasks to do so. Collaboration means that business and engineering collaborate about how best to build the product and what the product should do between iterations. They illustrated some important data about the application of Scrum and XP in two projects. They

illustrated briefly the use of Scrum to manage the steps taken to develop software, in conjunction with the use of XP practices to ensure the quality of the software.

- Bashir and Qureshi proposed a framework that is based on a combination of the strengths of Rational Unified Process (RUP), XP and SCRUM to be used for small, medium and large projects [17]. Their framework focused on modifying RUP using XP and scrum. 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 discipline that acts as a skeleton for all kinds of projects, especially for large scale but doesn't give best engineering practices to achieve simplicity, reliability, and quick adaptation to changing requirements. Scrum provides the best mechanism to manage and track the application development but lacks structured approach and best engineering practices. XP has best engineering practices, but lacks structured and formalized nature as well as lack of best managerial aspects. I find that the proposed framework provides a preliminary idea of merging the three approaches, but it doesn't achieve the agility because it tends to RUP.
- Malhotra and Chug proposed "IXSCRUM" which is a framework that integrates Scrum and XP [3]. This model was based on merging the processes of Scrum and XP to achieve higher quality. This model has the engineering approach of XP along with the management approach of Scrum. The authors tried to ensure the suitability of this approach by applying it on a Shopping application which involves buying products from the categories like hardware goods, software goods etc. HP's Quick Test Professional was chosen as a testing tool. I find that this merging isn't practical and need a lot of efforts to be executed. In other words, the model doesn't include guidelines or details needed for applying the model.

Finally, most of the efforts that were made to combine Scrum and XP is not enough because most of them don't introduce a guide that includes the activities and steps of

applying their combination. In addition, some efforts decrease the agility. Therefore, this paper focuses on introducing an enhanced Scrum framework that includes guidelines for applying it.

5- The Stages of Preparing the Enhanced Scrum Framework

This paper proposes an enhanced Scrum framework that combines some XP practices into Scrum framework activities to enhance it. The enhanced Scrum framework includes a set of elaborated guidelines that are required to achieve each Scrum activity. Then, these guidelines are validated to reach the final enhanced Scrum framework. The stages of preparing the enhanced Scrum framework are shown in figure (2) and will be explained in the following subsections. The stages are:

- 1- Preparing the outline of the enhanced Scrum framework.
- 2- Elaborating a preliminary set of guidelines.
- 3- Validating the guidelines.
- 4- Finalizing the guidelines.

5-1 Preparing the Outline of the Enhanced Scrum Framework

The researchers studied many attempts of combining Scrum and XP and quality improvements of software projects that were based on Scrum, XP, or both. Also, the researchers studied in detail the features of Scrum and XP. The main sources of this stage are papers, books, technical reports, and opinions of experts published on the web sites. At the end of this stage, the researchers concluded that the Scrum framework activities can be enhanced by a set of XP practices.

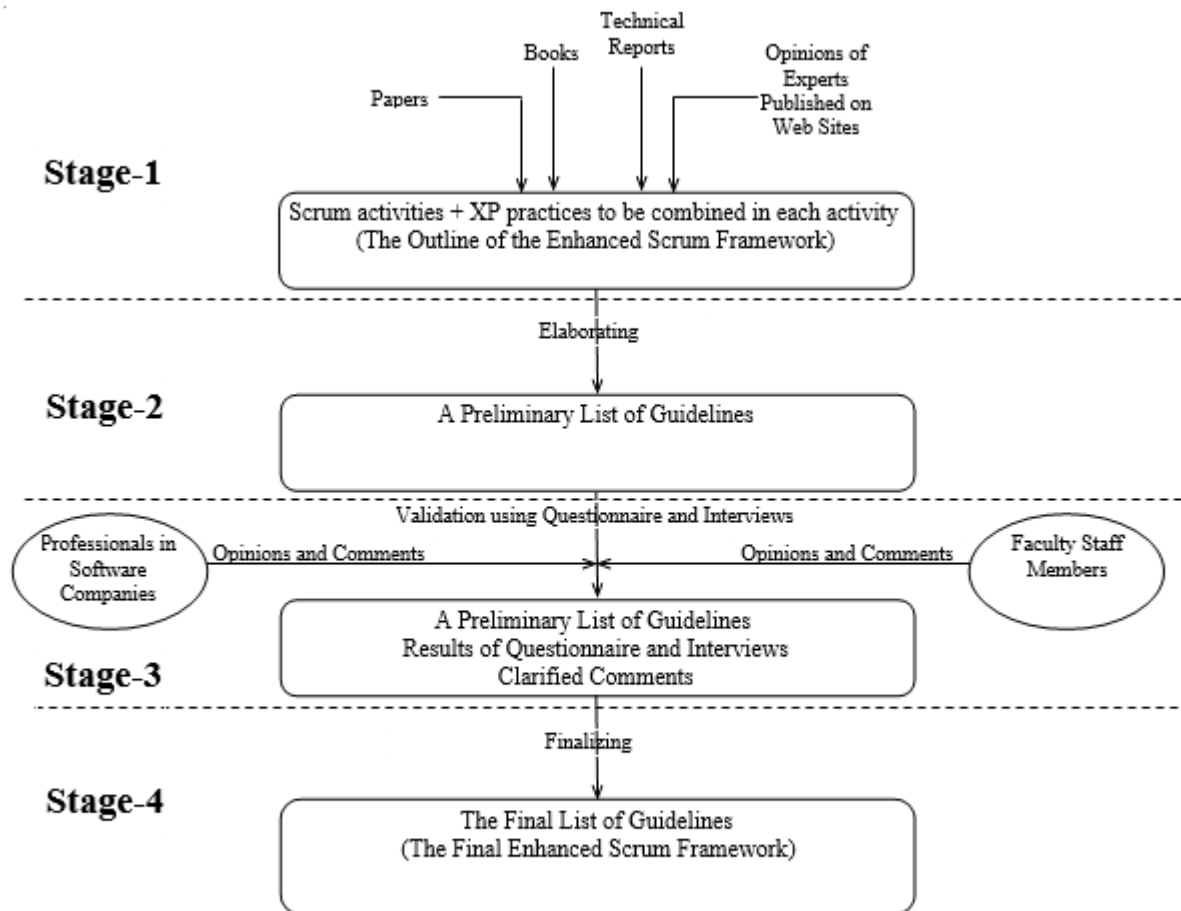


Figure (2): The Stages of Preparing the Enhanced Scrum Framework.

Table (3) illustrates the Scrum activities and XP practices to be combined in each activity. In table (3), the researchers used the Scrum framework activities that mentioned at the beginning of section (2). These activities are: preparing product backlog, sprint planning meeting and preparing sprint backlog, sprint, daily Scrum, sprint review and presenting an increment, and sprint retrospective. XP practices that should be combined with each Scrum activity are shown in the table. This table is the base of the enhanced Scrum framework that shows the cooperative relationship

between Scrum and XP. In addition, this table represents the outline of the enhanced Scrum framework. In the following subsections, more efforts will be done to complete the details of this outline framework.

Table (3): Scrum Activities and XP Practices.

Scrum Activity	XP Practices to be Combined in the Activity
Preparing Product Backlog	Simple Design
Sprint Planning Meeting and Preparing Sprint Backlog	- Simple Design - User Stories (it is XP feature)
Sprint	- Testing - Pair Programming - Coding Standards
Daily Scrum	-
Sprint Review and Presenting an Increment	- Refactoring - Continuous Integration - Collective Code Ownership
Sprint Retrospective	-

5-2 Elaborating a Preliminary Set of Guidelines

The researchers elaborated a preliminary set of guidelines for achieving each activity in the Scrum framework taking into consideration XP practices that increase the quality of performing this activity. Table (4) illustrates a sample of the guidelines for the activity titled “Sprint Planning Meeting and Preparing Sprint Backlog”. During this activity, the XP practice titled "simple design" can be used in planning the sprint goal. In general, the researchers consider the simplicity and understandability during the formation of the guidelines of all Scrum activities.

Table (4): Sample of the Preliminary Guidelines of a Scrum Activity.

Scrum Activity	Guidelines
Sprint Planning Meeting and Preparing Sprint Backlog	1. The Scrum master calls a sprint planning meeting that must be attended by the whole team and the product owner. 2. The attendees decide the sprint duration that may range from one week to one month. The maturity of the team is one of the factors that are considered when taking this decision. 3. The product owner and development team identify a sprint goal and agree on it. 4. The team and Scrum master select a section of items/features from the top of the product backlog that can be achieved in the sprint duration. 5. The product owner presents each item and explains how it is working from a functional perspective. The whole team discusses the item in detail. The product owner can use the concept of writing 'User Stories' which are used in XP. 6. The Scrum master and the team estimate and calculate the team's sprint budget. This is the available number of hours the team has to actually work on the sprint. 7. The Scrum master and the team break the requirements into tasks that are necessary to make the product backlog item complete and potentially shippable within the sprint. The Scrum master and the team agree on a definition of "done", so everyone is aware what will have to be completed and included in the estimates. 8. The team and Scrum master must keep tasks small and estimate all tasks in hours. Ideally task estimates should be no more than one day. 9. The team and Scrum master add up all the task estimates for the selected product backlog items. If they are significantly over the team's sprint budget, the number of selected product backlog items must be reduced. 10. The team should commit to delivering the sprint backlog.

5-3 Validating the Guidelines

The researchers planned to validate the elaborated guidelines using a questionnaire that must be answered by a set of 60 experts and specialists. The respondents were originally classified into two groups equal in the number of members; where the first group includes 30 faculty staff members who are interested in software projects, agile methods, and quality assurance. The second group includes 30 professionals who work in software companies that have some attempts of using agile methods in their software projects. The actual number of respondents who completed the questionnaires were 40; 21 from the first group and 19 from the second group.

For the purpose of validation, the researchers designed a simple questionnaire as shown in figure (3) that gives the respondent a space to set his comments on each guideline. Then, the researchers sent the questionnaire form by email to the two groups. The researchers showed in the email the purposes of the questionnaire and determined two weeks for answering it and resend. After collecting the responses, the researchers held an interview with each respondent individually to discuss his comment on each guideline to reveal the ambiguity and to reach to a clarified comment.

Figure (3): Sample of the Questionnaire - Validation of the Guidelines.

Important Notes:		
• If you agree on the guideline, please put (✓) on OK • If you don't agree on the guideline, please put (✓) on NO and give your comment below • If you see that the guideline must be modified, please put (✓) on MODIFY and give your comment below		
Scrum Activity	Guidelines	Opinion and Comments
Sprint Planning Meeting and Preparing Sprint Backlog	1. The Scrum master calls a sprint planning meeting that must be attended by the whole team and the product owner.	☐ OK ☐ NO ☐ MODIFY Comments:
	2. The attendees decide the sprint duration that may range from one week to one month. The maturity of the team is one of the factors that are considered when taking this decision.	☐ OK ☐ NO ☐ MODIFY Comments:
	3. The product owner and development team identify a sprint goal and agree on it.	☐ OK ☐ NO ☐ MODIFY Comments:
	4. ...	
	5. ...	

5-4 Finalizing the Guidelines

After reaching the clarified comment on each guideline, the researchers studied it carefully to form the final one. In section (6), a final list of guidelines will be presented for each Scrum activity. Some important results extracted from the questionnaire are valuable to be stated in the following:

- 61 % of the guidelines had the answer “OK” and they didn’t need any modifications.
- 8 % of the guidelines had the answer “NO” and they were eliminated.
- 31 % of the guidelines had the answer “MODIFY” and they were subjected to modifications. Some modifications were done to reveal the ambiguity and the confusion in meaning. Other modifications were done to rearrange the sequence of some guidelines. Also, there were some modifications that were done to add or remove guidelines.

6- The Enhanced Scrum Framework

The enhanced Scrum framework combines some XP practices into Scrum activities to enhance them. Also, it includes a set of guidelines that are required to achieve each Scrum activity. In the following subsections, a final list of guidelines will be presented for each Scrum activity.

6-1 Preparing Product Backlog

In this activity, XP practice titled "simple design" can be used during determining system requirements and priorities. The product owner must emphasize on designing only what was needed to support the functionality being implemented. The guidelines that can be used to achieve this activity are:

1. The product owner collects the features that are required in the new product. These features are collected from users, customers, and other stakeholders. These features may represent whole project, fixing bugs, and/or enhancements.
2. Ensuring that the items in the product backlog are expressed in business terms, including functional and non-functional requirements.
3. The product owner identifies an initial list of the required features.
4. The product owner validates the required features and solve conflicts if found.
5. The Scrum master discusses the product backlog with the product owner.
6. The product owner prioritizes the features and prepares an ordered list of features which is known "product backlog".
7. The Scrum master provides high-level initial estimates for the product backlog items using points system such as Fibonacci numbers which are a sequence of numbers where each number is the sum of the previous two. Fibonacci numbers are 1, 2, 3, 5, 8, 13, 21 ... etc. The Scrum master can use these numbers for indicating the size of each item in the product backlog according to the item complexity.
8. The Scrum team and the Scrum master should participate in the estimation process and then negotiate the size of each backlog item as a team.
9. Once the backlog size is estimated, the Scrum master can ask the product owner to review the priorities. The product owner can see the relative size of the features and he/she might change these priorities.

6-2 Sprint Planning Meeting and Preparing Sprint Backlog

In this activity, XP practice titled "simple design" can be used during planning out the sprint's goals. The guidelines that can be used to achieve this activity are:

1. The Scrum master calls a sprint planning meeting that must be attended by the whole team and the product owner.
2. The attendees decide the sprint duration that may range from one week to one month. The maturity of the team is one of the factors that are considered when taking this decision.
3. The product owner and development team identify a sprint goal and agree on it.
4. The team and Scrum master select a section of items/features from the top of the product backlog that can be achieved in the sprint duration.
5. The product owner presents each item and explains how it is working from a functional perspective. The whole team discusses the item in detail. The product owner can use the concept of writing 'User Stories' which are used in XP.
6. The Scrum master and the team estimate and calculate the team's sprint budget. This is the available number of hours the team has to actually work on the sprint.
7. The Scrum master and the team break the requirements into tasks that are necessary to make the product backlog item complete and potentially shippable within the sprint. The Scrum master and the team agree on a definition of "done", so everyone is aware what will have to be completed and included in the estimates.
8. The team and Scrum master must keep tasks small and estimate all tasks in hours. Ideally task estimates should be no more than one day.
9. The team and Scrum master add up all the task estimates for the selected product backlog items. If they are significantly over the team's sprint budget, the number of selected product backlog items must be reduced.
10. The team should commit to delivering the sprint backlog.

6-3 Sprint

In this activity, the Scrum team can use XP practices: Testing, Pair Programming, and Coding Standards. Unit tests are written prior to the construction of code. This practice forces developers to understand the interface and the expected functionality. The tests accumulate over the duration of a project, providing a library of regression tests. Once programming is started, all the tasks would be accomplished through pair programming conforming to coding standards. Although Scrum does not really prescribe how we should go about delivering the tasks in the sprint, the researchers proposes some important guidelines that can be used to achieve this activity.

1. The Scrum master must provide the team with a suitable environment for work. During Scrum, the Scrum master must provide them with support, guidance, coaching and assistance.
2. The Scrum master and the team must take into consideration that the sprint duration is fixed. They can add tasks if they discover the necessity of these tasks. However, additions to scope should be offset by compensating reductions in scope.
3. Ensuring that the concept of "Done" is clear to all team members. The team must complete one feature at a time before moving on to the next.
4. Testing is integrated throughout the lifecycle.
5. The Scrum master protects the team from any interference during the sprint. Ideally, once a Scrum team has committed to a sprint, they should be left to focus on delivering what they've committed to.
6. The Scrum master and the team track the progress using daily burndown chart and daily Scrum meeting.

6-4 Daily Scrum

The Scrum master holds a daily stand-up meeting with the team and the product owner. The guidelines that can be used to achieve this activity are:

1. The Scrum master and the team hold a daily routine meeting that must be in the same place and at the same time
2. The Scrum master facilitates the Scrum meeting and keeping it focused, timely, and 'on topic'.
3. The team stands in a half circle around their sprint whiteboard.
4. Each team member reports back to the team in turn. Only the person reporting back should speak at one time. The report should be concise and focused.
5. The report should address three key questions about what did they do yesterday, what will they do today, and what's in their way.
6. If an issue needs further discussion, the Scrum master avoids discussing it in detail until finishing the Scrum meeting. Only the members who have this issue can stay to complete their discussion.

6-5 Sprint Review and Presenting an Increment

In this activity, the Scrum team can use XP practices: refactoring, continuous integration, and collective code ownership. When the sprint finishes and the release is evaluated in the sprint review, any refactoring of the system requirements can be added to the backlog before a new sprint begins. Refactoring allows for the incremental improvement of the design and class structure to support new functionality. The Increment is the total of all the product backlog items accomplished during the current sprint and all previous sprints. At the end of each sprint, the product owner will receive a deliverable, and will be able to see the

incremental growing of the product [7]. The guidelines that can be used to achieve this activity are:

1. The Scrum master ensures that the team has followed the concept "Done". They should 100% complete each feature before moving on to the next. The completed features will form a perfectly shippable product.
2. The team members review what they've achieved and demonstrate their contribution to the product.
3. The team and Scrum master show all key stakeholders what's been done and ask them to provide feedback.
4. The team and Scrum master identify the changes and document them.
5. The team and Scrum master must ensure that the software is always in a shippable state.
6. The team restructures the system without changing its behaviour to remove duplication, improve communication, simplify, or add flexibility [18].
7. The team must ensure that all changes are integrated and tested every few hours, or at least daily in the worst case [4].
8. All tests are run, and they must be passed for accepting the changes in the code. The team integrates and builds the software multiple times per day.
9. Once code and its associated tests are checked into the code base, the code can be altered by any team member.

6-6 Sprint Retrospective

The sprint retrospective meeting is useful for the Scrum team to inspect itself and identify the possible improvements to be considered during the upcoming sprint. The guidelines that can be used to achieve this activity are:

1. The Scrum master invite the development team and product owner to the sprint retrospective meeting.
2. Review the final Burndown Chart and answer the questions: How did it go? Did the team deliver what they committed to at the start of the Sprint?
3. Review and track the team's velocity. Velocity is the number of points estimated on the original product backlog for the 100% completed items in the sprint. Velocity can be used in sprint planning as a gauge for how much the team could realistically achieve.
4. Discuss what went well and document it to make sure it's repeated next time.
5. Discuss what could have gone better and document it to understand why.
6. Decide what the team will do differently in the next sprint. In other words, they pick a few actionable points that can actually be done differently in the next sprint.

7- Conclusion

This paper aimed to propose an enhanced Scrum framework that combines some practices of XP approach in Scrum framework to produce quality software on time. Therefore, the researchers studied many attempts of combining Scrum and XP and quality improvements of software projects that were based on Scrum, XP, or both. Also, the researchers studied in detail the features of Scrum and XP.

Then, the researchers proposed an enhanced Scrum framework that is based on Scrum activities and some XP practices and features. The Scrum framework activities are: preparing product backlog, sprint planning meeting and preparing sprint backlog, sprint, daily Scrum, sprint review and presenting an increment, and sprint retrospective. XP practices that should be combined with Scrum activities are: Simple Design, Testing, Pair Programming, Coding Standards, Refactoring, Continuous Integration, and Collective Code Ownership. Also, the XP feature titled

“User Stories” that can be used during the activity titled “Sprint Planning Meeting and Preparing Sprint Backlog”. To make the enhanced Scrum framework more applicable than many previous attempts in this domain, the researchers elaborated a set of guidelines for achieving each Scrum activity. The enhanced Scrum framework was validated by a group of 40 experts and specialists in software projects.

The researchers concludes that the cooperative relationship between Scrum activities and some XP practices, which is exploited in the enhanced Scrum framework, can help the Scrum team to produce software with higher quality. In addition, the researchers concludes that more attention must be given to the combinations of agile methods.

8- Future Work

There are many issues and ideas can be tackled in future. Most of these issues and ideas are related to the domain of agile methods, such as:

- Using Scrum framework to achieve higher levels of Capability Maturity Model Integration (CMMI) for software companies.
- Studying the relationships among XP practices.
- Using the enhanced Scrum framework to improve the quality of web sites.
- Extending the enhanced Scrum framework to be applied to large-scale projects.
- Proposing a quality assurance model for software projects based on the enhanced Scrum framework.

References

- [1] A. Qumer and B. Henderson-Sellers, "An Evaluation of the Degree of Agility in Six Agile Methods and its Applicability for Method Engineering", Information and Software Technology Vol. 50 Issue 4, 2008, pages 280–295, 2008.
- [2] Artem Marchenko and Pekka Abrahamsson, "Scrum in a Multiproject Environment: An Ethnographically-Inspired Case Study on the Adoption Challenges", Agile, 2008. AGILE '08. Conference, IEEE, Date 4-8 Aug. 2008.
- [3] Chhavi Malhotra and Anuradha Chug, "IXSCRUM-A Framework Combining Scrum and XP", International Journal of Scientific & Engineering Research, Vol. 4, Issue 7, July-2013.
- [4] Dean Liffingwell, "Scaling Software Agility – Best Practices for Large Enterprises", The Agile Software Development Series, Pearson Education Inc., 2007.
- [5] Giulio Concas, Marco Di Francesco, Michele Marchesi, Roberta Quaresima, and Sandro Pinna, "An Agile Development Process and Its Assessment Using Quantitative Object-Oriented Metrics", 9th International Conference, XP 2008, Limerick, Ireland, Proceedings, June 2008.
- [6] http://www.scrumalliance.org/pages/what_is_scrum, Last Visit: 14 February 2014.
- [7] Ionel Năftănăilă, "Critical Analysis of the Scrum Project Management Methodology", Annals of the University of Oradea, Economic Science Series, Vol. 17 Issue 4, December 2008.
- [8] Jeffrey A. Livermore, "Factors that Significantly Impact the Implementation of an Agile Software Development Methodology", Journal of Software, Vol. 3, No. 4, APRIL 2008.
- [9] Jorge Edison Lascano, "eXtreme Programming may be embedded inside Scrum", Systems & Software Technology Conference, SSTC-2009, www.ieee-stc.org/proceedings/2009/pdfs/sps81.pdf, Last Visit: 12 February 2014.
- [10] K. Beck, "Extreme Programming Explained: Embrace Change", Addison Wesley, 1999.
- [11] Kane Mar and Ken Schwaber, "Scrum with XP", InformIT, Mar 22, 2002, <http://www.informit.com/articles/article.aspx?p=26057>, Last visit: 22 February 2014.
- [12] Karlheinz Kautz and Sabine Zumpe, "Just Enough Structure at the Edge of Chaos: Agile Information System Development in Practice", 9th International Conference, XP 2008, Limerick, Ireland, Proceedings, June 2008.
- [13] Ken Schwaber and Jeff Sutherland, "The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game", Scrum.Org, October 2011.
- [14] Kenneth S. Rubin, "Essential Scrum: A Practical Guide to the Most Popular Agile Process", Addison-Wesley, Pearson Education, Inc., 2013.

-
- [15] Laurie Williams, "A Survey of Agile Development Methodologies", 2007, agile.csc.ncsu.edu/SEMaterials/AgileMethods.pdf, Last Visit: 26 February 2014.
- [16] M. Angioni, D. Carboni, S. Pinna, R. Sanna, N. Serra, A. Soro, "Integrating XP Project Management in Development Environments", *Journal of Systems Architecture* Vol. 52, Issue 11, pp. 619–626, 2006.
- [17] M. Salman Bashir and M. Rizwan Jameel Qureshi, "Hybrid Software Development Approach for Small to Medium Scale Projects: RUP, XP & Scrum", *Sci. Int., Lahore*, 24(4), 2012.
- [18] Mark C. Paulk, "Extreme Programming from a CMM Perspective", *IEEE SOFTWARE*, November/December 2001.
- [19] Markus Rittenbruch, Gregor McEwan, Nigel Ward, Tim Mansfield, and Dominik Bartenstein, "Extreme Participation - Moving Extreme Programming towards Participatory Design", *Proceedings of the Participatory Design Conference*, Malmo, Sweden, pp.29-41, June, 23-25, 2002.
- [20] Naga Sri Morampudi and Gaurav Raj, "Evaluating Strengths and Weaknesses of Agile Scrum Framework using Knowledge Management", *International Journal of Computer Applications*, Vol. 65 No. 23, March 2013.
- [21] Nagy Ramadan Darwish, "Improving the Quality of Applying eXtreme Programming (XP) Approach", *International Journal of Computer Science and Information Security (IJCSIS)*, Vol. 9 No. 11, November 2011.
- [22] Nagy Ramadan Darwish, "Towards an Approach for Evaluating the Implementation of eXtreme Programming Practices", *International Journal of Intelligent Computing and Information Sciences (IJICIS)*, Ain Shams University, Vol. 13 No. 3, July 2013.
- [23] P. Deemer, G. Benefield, C. Larman and B. Vodde, "The SCRUM Primer", Ver. 1.2, 2010. <http://www.brianidavidson.com/agile/docs/scrump1mer121.pdf>, Last Visit: 5 March 2014.
- [24] Richard Moore, Kelly Reff, James Graham, and Brian Hackerson, "Scrum at a Fortune 500 Manufacturing Company", *AGILE 2007*, Computer Society, IEEE, 2007.
- [25] Soumyadipta Paul and K. John Singh, "Be Agile: Project Development with Scrum Framework", *Journal of Theoretical and Applied Information Technology*, Vol. 40 No.1, 15 June 2012.
- [26] VersionOne Inc., "The State of Agile Development - 3rd Annual Survey: 2008", http://www.versionone.com/pdf/3rdannualstateofagile_fulldatareport.pdf, Last Visit: 8 March 2014.
-