

Agile Programming - Scrum XL

Prema.S.Thomas
Assistant Professor
M.C.A Department
Rajagiri College
Kalamasserry, Kochi

Satheesh Thomas
Managing Director
Pumex Infotech Pvt Ltd
Inforpark, Kochi

ABSTRACT

This paper presents a case study on the procedures and practices of the new approach to agile programming which is named as scrum XL. Need for a new approach was identified in my earlier paper called "Agile Programming-A New Approach"[1]. The approaches mentioned in this paper were implemented in a real working environment. It explains the features that have been successfully put into practice. The strategy adopted for implementing the features is discussed. It gives a detailed report on the steps and results of the features that are stated.

IndexTerms— Bugzilla, Backlog, Sprint, Scrum, Scrum XL

1. INTRODUCTION

Agile methods emerged as new generation Software development lifecycle methods in recent past. In the current technology world where technology and requirements are changing very frequently, these methods are more relevant over traditional methods. There are multiple agile methods existing as software development lifecycle methods. Scrum is a popularly used agile methodology and is adopted by many companies in last decade. Existing agile methods and its drawbacks are explained in detail in my earlier paper called

"Agile Programming-A New Approach"[1] and as a result a modified approach was found necessary to overcome the drawbacks.

This paper is intended to state the new methodology and an example of its practical implementation in a real project scenario.

The drawbacks of the existing agile programming methods are :

- Doesn't support projects of large size (more than 10-20 people)
- Not suited for multi site projects
- Unable to do overall optimization on schedule using fast tracking

The new approach addresses these issues and finds a solution to these problems.

2. METHODOLOGY

The technique which is applied to overcome the above mentioned issues are addressed in the new methodology which is Scrum XL. The first and foremost thing to be done is to identify the tool in which you can map the features. There are many standard tools such as Bugzilla, Mantis etc..

Once the environment is identified we need to put all the features to this common platform based on the initial customer input. This will be marked as unspecified version which can also be called as the product backlog.

The next activity is to involve the customer in a meeting with the development manager and the development team in order to discuss the priorities and plan it against a release version of the product. It will be specified only after the discussions when the specifications have been prioritized. The listed specifications will be the product backlog. **Figure 1.1** shows relationship between the various teams involved in putting all the features to a common platform like bugzilla. The output of the meeting is a product backlog which serves as input for the sprint which is a feature list against an agreed version.

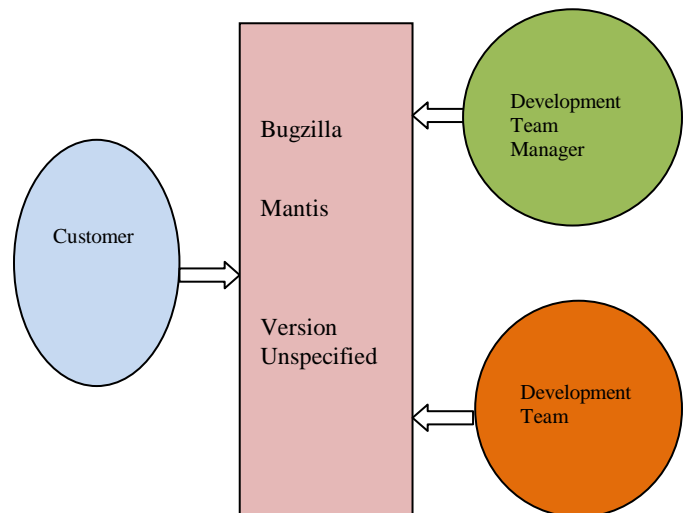


Figure 1.1 – Environment

The outcome of the meeting is a list of features of the system to be developed against an agreed version number. Feature list against an agreed version is equivalent to a sprint. As mentioned in the paper- Agile Programming-A New Approach", a sprint is the basis of agile programming. The most critical features are taken into account and implemented first. This method of identifying the high priority specification leads to the first version of the product say *product 1.0*. Then the next high priority specification is taken and implemented as a sprint. The requirements that are finalized as the first version of the product cannot be changed under any circumstances. Once the versions are finalized then an estimate of each of the versions is made. Estimation is done for the time as well as the budget for the sprint. The customer and the development manager finalizes on the budget and time frame for each sprint. The team then focuses on the execution of the sprint for the release of the first version . As mentioned no change in either the schedule or the budget can be made after the product version is finalized. The changes can be incorporated in the next versions if needed. In the development process all other practices for scrum can be applied. The steps involved in the sprint definition process are as follows:

- All the features will be listed as unspecified initially which is equivalent to product backlog
- Identifying a Sprint ie prioritized features against a version
- Prioritize the requirements and move it to a particular version say 1.0,1.1,1.2 etc..
- Estimate for the next immediate version say 1.0 and deciding on the budget and schedule
- Team focuses on execution of that approved version
- No change in the allowed budget or schedule
- All procedures of scrum is applicable
- The team focus on development as per the development process of the organization

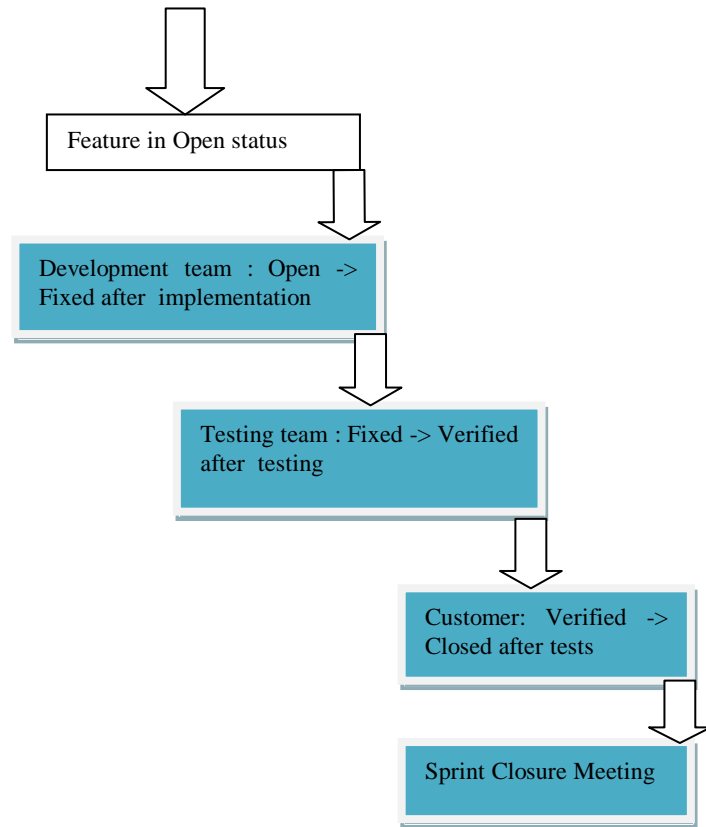


Figure 1.2 –Sprint development flow

Any bug reported during the Sprint also will be reported against the same version and the same set of procedures will be repeated for resolution.

3. NORMAL SCRUM

In the case of a normal scrum the requirements are taken from the product backlog and executed after the sprint is identified. This procedure will be executed in a cycle till the complete development of the product. **Figure 1.3** given below shows the execution process in the case of a normal scrum.

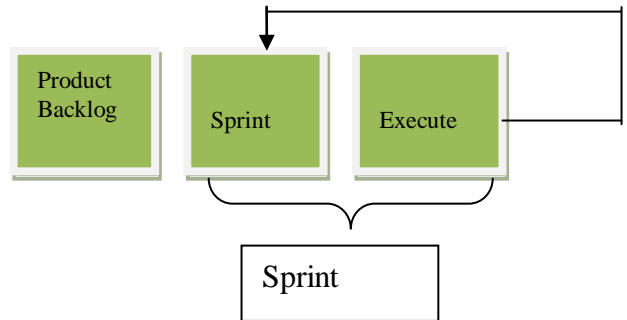


Figure 1.3 –Normal Scrum cycle

4. MAPPING TO THE NEW METHOD

Let us now map the process from the normal scrum to the new methodology. With the new methodology the sprint definition will consist of a product backlog from which the sprint has to be defined after a meeting of the customer, development manager and the development team. Initially the requirements without the prioritization is the unspecified version of the product. The meeting focuses on prioritizing the requirements and then assigning a version for the product to be developed. This version specified becomes the sprint definition and a sprint backlog is created out of it. The result being a versionized list of the requirements.

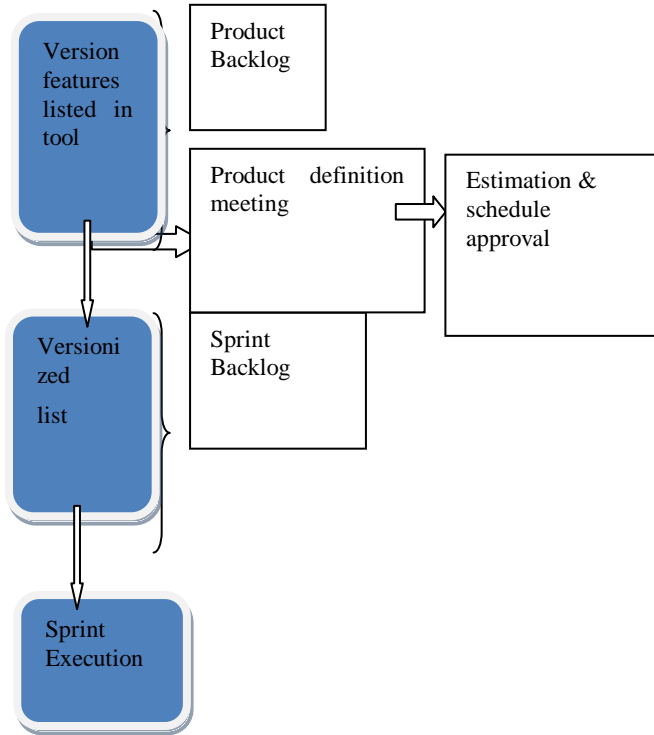


Figure 1.4 –Sprint Definition process

The new process for the sprint execution would consider the requirements from the version and then change the status to open. Then the requirements are fixed after the meeting, if not fixed then again it is treated as open and so reviewed and then fixed. Once the development process is over for the sprint, the tester would test the output and is found correct then it is closed otherwise again it is modified and send for verification by the tester till it is closed. The requirement is closed after verification by the customer that the needs are met. Once the sprint definition is finalized the execution of the sprint can be done in the regular way through stand up sprint meetings, sprint backlog etc.. **Figure 1.5** depicts this iterative process that happens in the sprint execution.

The result of the new approach is a list of reports that could be generated like

- Sprint Definition
- Progress of the Sprint
- Status of the features
- Comments on how the bug could be resolved
- History of the bug like who reported and how was it resolved
- Time taken to correct the bug
- Severity of the bugs etc..

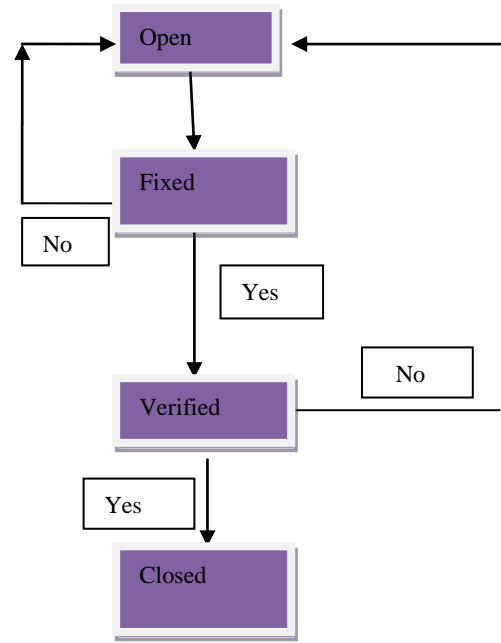


Figure 1.5 – Sprint execution flow

Summary of Scrum Vs Scrum XL

SI No	Normal Scrum	Scrum XL
1	Product Backlog	List of features in unspecified
2	Sprint backlog	Feature list in selected version
3	Sprint planning meeting	Sprint planning meeting
4	Scrum Roles	Same
5	Sprint planning meeting	Same
6	Daily Standup meeting	Same
7	Sprint duration (1-4 weeks)	Typically 3-16 weeks
8	Progress reports	Available as bug status online
9	Multisite difficult	Possible with all info available online
10	Sprint closure meeting	Same reviewed online

5. EXAMPLES

Let us now take a typical example of a product and see how the new method could be applied to the development of the product. The development process first assumes a common tool such as bugzilla and maps the features to this common tool. Thereafter the features are listed as per the definition stated in **Figure 1.4**. As discussed the versionized list is then generated after the meeting of the development team and the customer along with the development manager (Sprint Planning meeting). The features mapped to the common platform will look like the screen shot given below. The initial state of the product backlog is an open state where all the features are listed. It is from this list that the versionized list is generated. **Figure 1.6** shows the list after mapped a specified version to bugzilla.

		Version		
		6.3.2	unspecified	Total
Status	NEW	18	48	66
	REOPENED	.	3	3
	RESOLVED	.	9	9
	VERIFIED	3	3	6
	CLOSED	.	12	12
	Total	21	75	96

Figure 1.6

The table given above shows the version id form of the product YasooYamoo. Then the estimation will be done for the immediate version say 6.3.2 and team will proceed with the implementation. Other features will remain under unspecified.

The features in 6.3.2 will be addressed by the development team and the status will change from Open-> Fixed->Varified-> to close as shown in the process of sprint development process (as in **Figure 1.2**)

Another report that lists the people assigned to a particular product. Similarly there are quite a number of useful reports that could be generated after mapping it to bugzilla.

Figure 1.7 shown below shows all the releases (or Sprint) against its status . That will give an overall view of the entire product cycle.

	NEW	REOPENED	RESOLVED	VERIFIED	CLOSED	Total
6.1	.	1	.	12	259	272
6.1-new	.	.	.	4	6	10
6.1.1	.	.	.	59	25	84
6.2	.	.	.	10	2	12
6.2.1	.	.	.	64	.	64
6.3	.	.	.	14	2	16
6.3.1	8	.	.	5	.	13
6.3.2	18	.	.	3	.	21
phase5	.	1	.	.	60	61
phase6	15	15
support	2	2
unspecified	48	3	9	3	12	75
Total	76	5	9	174	381	645

Figure 1.7 : Overall Status

Reports of any combination can be generated as per the requirement. All the people involved can also see the status, comments and history with a specified login and password. The history and comments will be displayed with audit log and time stamp.

Choosing any parameter listed in the tool helps to generate the corresponding reports of the product.

All the people in the system can also see the progress of each feature and also add comments. These comments will be available for future reference.

The system also helps in generating these reports which can minimize the tracking and progress reporting efforts.

6. CONCLUSION

The new methodology focuses on the three major issues stated in the existing agile programming method. With the introduction of Scrum XL the disadvantages were overcome. Mainly on the issue with the inability to support large projects and also multi sited projects. This new method can handle projects that are scattered around the globe. The people working for the product can log on to the system from any part of the world and then work on it. The size of work group is also not a problem. It can handle more than 20 people. All agreed data is online for access to the people involved. The effort and schedule estimation to be done outside

the tool for a selected version (or Sprint) and the optimization can be done to meet the stakeholder requirements.

In this way Scrum XL overcomes the limitations of Scrum and at the same time work on the basic Scrum methodology.

REFERENCES

- [1] Prema.S.Thomas, Agile Programming-A New Approach
- [2] David Berger, Bill Cloke, Lean products start with Lean design.
- [3] James P. Womack, Daniel T. Jones et. Al (1990), The Machine That Changed the World: The Story of Lean Production.
- [4] Jones, D.T. & Womack, J.P. (1996). Lean Thinking: Banish Waste and Create Wealth in Your Corporation,
- [5] Ian Sommerville, Software Engineering, Pearson Education, Eight Edition
- [6] Beck, K., Extreme Programming Explained, Reading, MA: Addison-Wesley Longman, Inc., 2000.