

# Study Of User Story Tools That Are Beneficial To Agile Software Development

Ankur Srivastava, Premveer, Arvinder Kaur  
Department of IT,USICT,GGSIPU

## Abstract

*We all know that requirements' gathering is very crucial step of software development process. Since requirements gathered need to be correct and consistent, a proper process need to be used. Not only we need a proper process but in a environment where requirements may change or vary considerably with time and the software is developed as a product in consequent releases like in a agile software development, a proper communication channel is also required among different people dealing with the software. These can be either belonging to software development team or users of the software. That is why a tool is needed to fill the gap between these people so that each and every requirement of the product can be covered effectively and each team member of developing team can have or view the changed requirements instantly without taking too much time. In this paper, the author has tried to review various user story tools available in the market. For this purpose, the author has selected two user story tools namely Planbox and Scrumninja which are available in the market for one month trial and tried to implement them through a software project and compare their performance based on multiple parameters. The reminder of this paper is organized as follows. In Section I, an introduction of Agile software development is given. In Section II, an effort has been made to introduce User Story. Section III focuses on Scrum which is a popular agile software development technique for which both the user story tools are designed. In Section IV, Planbox tool is discussed thoroughly. In section V, Scrumninja tool is introduced and its various features are discussed. In Section VI the author has compared both the tools citing their advantages and disadvantages. Finally Section VII is the concluding remarks.*

**Keywords-Agile Software Development, User Story, Planbox,Scrumninja,Scrum**

## 1. Introduction

Agile software development is a development technique or methodology in which a group of software development methods are used based on iterative and incremental development. Requirements and solutions are gathered through collaboration between self-organizing different cross-functional teams. This collaboration works for the improvement in software quality and also increases the development team sensitivity towards changing customer requirements. We focus on frequent "releases" in short development cycles called iterations which aims at improving productivity and one can also introduce requirements checkpoints where new customer requirements can be adjusted. Other elements of Agile software development include: pair programming, performing repeated code review, unit testing or module testing of the entire code, programming of features is not done till they are actually required or confirmed, a management structure which is simple and code transparency, anticipating changes in customer's requirements as time moves on and the better interpretation of the problem, and regular communication with the customer and among other programmers and team members [1]. Following are the advantages of this methodology:

1. Planning which is adaptable to circumstances.
2. Evolutionary development [2].
3. Quick delivery.
4. A time-bound incremental and iterative approach [3].
5. Rapidness and flexibility towards change [3].

Each iteration is like a miniature software project of its own that includes all of the tasks such as

planning, requirements analysis, design, coding, testing, and documentation to release the mini increment of new functionality.

## 2. What is User Story [4]?

As the name suggests 'Agile' time is very important factor. Thus we cannot delay the release of product due to change in requirement. Each member of the development team should be able to come to know of the changes required by each user in his/her requirements in a definite period of time. Thus a tool which we generally called as 'User Story tool' is used to gather and represent the requirements. It basically represents the product features in terms of user stories. We can split the features as functional tasks of the users that are concerned with the software. In software development, it is generally seen that it is advantageous to categorize requirements based on the user. So the tool should have this capability. A user story tool nowadays is available as a Agile Project Management tool or I can say they are integrated within the agile project management software. This is because the user story tool doesn't simply represent the requirements; it also provides a platform to manage the software being developed. A team member can check the status of various tasks, can also update his/her status and can also view the tasks to be completed in current and future iterations. Nowadays, due to distributed software development, it is not possible for the team members to interact directly with the users. So the user story tool must be supported by internet application through which team members and users can communicate with each other. This feature eliminates distance and time gap. Also since each member can communicate with each other member, this removes inconsistency at any stage of development.

A proposed user story tool can look like:

STORY NUMBER(10)		
STORY TITLE(STUDENT POINTS(5 Pts)		STORY STATUS(STARTED)
STARTING TIME( 25/05/2013 1300 hrs)	ESTIMATED TIME OF COMPLETION(5 hrs)	
-----STORY DESCRIPTION i.e. tasks----- ----- -----task1----- ----- -----task2----- ----- (Check the student info like name, address, dob, mobile no., course info etc.)		
UPDATE OPTION	COMMENTS OPTION	ATTACH FILE OPTION
PERSON RESPONSIBLE FOR THE STORY(i.e. Developer)		RISK LEVEL OF TECHNOLOGY USED ( 2)
DELIVERY CONTACT (PERSON NAME)		

**Figure-1**

Two tools are selected for the research. These are:

1. Planbox[5]
2. Scrumninja[6]

Both these tools are Scrum[7] based which is a popular agile software development technique and therefore next section discusses Scrum.

## 3. Scrum

Scrum is an iterative and incremental agile software development method for managing software projects and application development. Scrum has given the new direction and created new interest in project management. It has also challenged the conventional ideas about such management. Scrum targets project management principles where it is difficult to plan

ahead. Feedback loops are used in the main management technique against traditional command and control oriented management. It is a new approach for planning and managing projects and also helpful for all levels of decision-making. Scrum project management basically consists of a team that is made responsible for carrying out entire operations of the management.

A Scrum team primarily consists of following three members namely:

**Product Owner** [8]-The Product Owner represents the users and is the voice of the customer. It is the responsibility of his/her to make sure that the team gives importance to the business. The Product Owner writes user-specific tasks that is the user stories, prioritizes them, and adds them to the product backlog. Scrum teams must have one Product Owner. It is generally preferred that the product owner be a member of the development team.

**Development Team** [8]-The Development Team is responsible for delivering product increments at the end of each Sprint. A Development Team is made up of around 3-9 people. Each of whom has cross-functional skills which are required for analysis, design, development, testing, any technical communication and document, etc. The Development Team in Scrum is self-organizing.

**Scrum Master** [9]-Scrum Master is the head of the Scrum process. ScrumMaster makes sure that the team delivers maximum performance depending on its ability by removing obstacles that may come while implementing the tasks within the sprints to achieve goals and deliverables. The Scrum Master not only serves as a team leader, but also acts as a buffer between the team and any unwarranted circumstances. The Scrum Master makes it possible for the smooth execution of the Scrum process. The Scrum Master devise rules and regulations and often plays a key role to protect the Development Team and keep it focused on the tasks. The role can also be called as a servant-leader to reinforce these dual perspectives. Many a time, a Scrum Master is mistaken for a Project Manager. The difference is that, while the Project Manager may also have people management responsibilities in addition to the role of a Scrum Master, a Scrum Master shall not hold any such additional people responsibilities.

## 4. Planbox

Planbox has a plenty of features. It is a simple and easy to use agile project management software that helps you plan, collaborate and deliver [5]. Planbox is an agile software that follows the scrum method and its principles. Scrum is highly customer-driven and therefore features are usually translated into user stories in the scrum process [5]. A story explains how a feature is to be used and gives it context. A proper way to write stories is to start by: As a user or As a client. As a user it provides the facility to access one's data online and as a client it provides the facility to have a billboard printed. Stories are called items in Planbox and they are containers of tasks [5]. Some notable features of planbox are:

- Intuitive User Interface.
- Real-time Collaboration.
- Task based project management.
- Time Tracking.

Planbox introduces an intuitive UI your entire team can use. It gives the feeling of a virtual story board which is better viewable as compared to simple posts of stories. It provides the facility to drag & drop items and tasks to prioritize them according to the situation. It also serves as a filter for each member of the development team to see only the work he/she is involved in. Planbox allows you to collaborate in real-time with your entire team, whether you are distributed or 3ollocated. Item board is updated in real-time without any delay. Planbox allows team members to share files with other teammates and open them from Planbox. This sharing feature comes handy when team members are separated by geographical boundaries and are located in different cities or countries. Team members can also send comments so fast they can use it like a chat option. Real time activity feed let's you know what's going on in the current process of development. A team burndown chart can also be generated which is very helpful to monitor the work progress. There is an ideal process line and alarm line in the burndown chart. The former is initially estimated progress line without any risk considerations. The latter is initially estimated progress line with empirical risk estimations. At the end of iteration, actual progress result is marked on the chart for checking. The items shared between team members are also marked and described and their links to wiki can also be posted. Another feature of Planbox is task based project management. This feature makes Planbox perfect project management tool. Admin or team leader can manage and assign tasks online, drag and drop his

priorities. It also provides the facility to get visibility on the teams' status and progress for each and every team member. Each team member can mark items as blocked and can request help from the team. Multiple item statuses allow one's to review the team work. Another feature of Planbox is of time tracking. Team leader can organize the team based on their work capacity- unlock reporting features to get his/her team faster and more efficient. Team leader can estimate and plan the sprints. He/she can even get notified if someone is overbooked. He/she can log and track time spent on each task and can even generate time sheets at the touch of a button.

The author has tried to manage the project Student Info Automation System through planbox for one month trial and has find out various advantages of planbox. These are:

- Team leader can move the item from one sprint to another sprint to the desired iteration.
- He/she can move the item from the current sprint to the backlog.
- Burndown chart can be generated in terms of multiple parameters. These are:

Items, Item points, Estimate hours, Logged hours, Tasks, Business value

- Planbox has an option to compare the performance of different team members. It does so in terms of power level.
- A mobile phone app for Planbox is also available.
- It has a velocity meter to keep track of progress of work.

Only one disadvantage is there. There is no automatic division of backlog into sprints. We have to perform the division manually.

## 5. Scrumninja

One of the main tasks of the software project management team is to perform feature delivery and story acceptance. This goal is achieved by continuously tracking, communicating and sharing information that is available at any given time [6]. Also this information needs to be latest info. Scrumninja provides the facility to change things at a moment's notice. Scrumninja is a simple and easy to use scrum tool. It has the physical resemblance and functional intuitiveness of a real index card wall plus

key productivity features to make the scrum tool powerful such as automatic burndown chart plotting and history storage but not encumbered by feature bloats that ultimately serve to bog down work [6]. It provides the facility of virtual index card wall view. It makes it possible to have strong visual support. In order to keep team members focused so that they concentrate only on the relevant facts and items i.e. tasks, it is important to design a proper task card. It is also useful for referencing in daily standup meeting. Lets discuss some distinguished features of Scrumninja. It provides the facility to create, stack and arrange cards via drag and drop just like a realistic card wall [6]. Swim lanes reflect the status of tasks in the development process [6]. It is adaptable to any size team from small and medium to large. This tool can handle the complex projects quite well and backs a large number of cards. In respect of user story, the card wall displays the breakdown of each story into individual tasks. This feature is beneficial in process of discovery as well as it keeps the developers focused. Every task has a remaining-hours estimate attached to it, which is used to automatically calculate the burndown [6]. Also, it has the option to calculate burndowns using only story points. Scrumninja has limited advantages:

- Automatically divides the user story into multiple sprints taking the headache out of the development team.
- Prioritization of stories is easy, just drag and drop each story on the list to the desired order.

and lot's of disadvantages. These are:

- On dividing the backlog into sprints based on the average of previous sprints, the estimate is bigger than the sprint capacity. You have to update the sprint capacity manually.
- Scrumninja will automatically rearrange your sprints every time you change a priority and it can be confusing when you have a large backlog.
- The Burndown feature is only a simple hour tracking chart.
- We cannot move items from one sprint to another.
- No mobile phone app.

## 6. Comparison between Planbox and Scrumninja

**Table-1**

Tool → Attributes ↓	Planbox	Scrumninja
Change of iteration for an item	Yes	No
Shifting of item in current sprint to the backlog	Yes	No
Burndown chart based on multiple parameters	Yes	No
Team members performance evaluation	Yes	No
Mobile phone/Tablet app	Yes	No
Velocity meter	Yes	No
Automatic division of backlog into sprints	No	Yes
Prioritization of stories(drag and drop facility)	No	Yes
Ease of use with large backlog	Yes	No
Accuracy of estimated time	More	Less

## 7. Concluding Remarks

Nowadays there is huge craze for the project management software among the organizations which follow Agile Software development. Organizations want a twenty four hour monitoring system which continuously monitors the progress of tasks assigned to each team member. A wide range of Agile Project Management softwares are available in the market. There is a healthy competition among those softwares. Each software tries to fully accommodate the modern practices of agile software development like Scrum, Extreme Programming etc. The user story tool Planbox which is studied in this paper is very rich in its features and can easily compete with others in the same class. On the other hand, Scrumninja tool is lacking most of the features that an organization would be happy to have. So the author has found Planbox tool considerably better than the Scrumninja tool. After studying about both the tools and comparing them based on multiple parameters, it is clearly visible from the comparison table that Planbox excels but it lacks risk assessment feature. The risk assessment feature is very important feature. Suppose a user wants to change a particular requirement and he/she notifies the development team on Planbox portal. Then the development team analyzes the changed requirement, predicts its advantages and the risks associated with the change. Then there should be a mechanism to convey risk information to the user so that he should also clearly understand the risk associated that the development team has evaluated. Also depending on the risk the development team can select the correct technology. This problem can be solved by giving a risk assessment button on the user interface next to task label. An option of unit test description and user interface needs would be an added advantage. The author believes that these project management softwares are emerging in the business and its patterns and practices will evolve and mature in the future.

## 8. REFERENCES

- [1]<http://www.agile-extreme-programming-books.us/>
- [2] Kent Beck and C. Andres, Extreme Programming Explained, 2<sup>nd</sup> edition book Addison-Wesley, 2005.
- [3] "Manifesto for Agile Software Development", <http://agilemanifesto.org> June 2010.
- [4] "A feasible user story tool for Agile software development?" by Michael J Rees at School of Information Technology, Bond University, Australia IEEE 2002 .
- [5] <http://planbox.com>
- [6]<http://scrumninja.com>
- [7] "Usage and Perception of Agile Software Development in an Industrial Context" by Andrew Begel and Nachiappan Nagappan at Microsoft Research, IEEE 2007.
- [8] "Agile Project Management with Scrum" Microsoft Press.
- [9] Gauthier, Alexandre, "What is Scrum".
- [10] "Future Implementation and Integration of Agile methods in Software Development and Testing" by Woi Hin, Kee at Motorola Global Software Group Malaysia IEEE 2008.
- [11] "Software Project Management" by Walker Royce foreword by Barry Boehm Pearson education 1<sup>st</sup> edition 2000.
- [12] "Rapid Development" by Steve McConnell Microsoft Press 1<sup>st</sup> edition 2001.