

Design and Development of Agile/Scrum Dashboards

Jerin Jose Philip

Department of Computer Science and Engineering
Mangalam College of Engineering, Ettumanoor
Kottayam, India

Surya Babu

Department of Computer Science and Engineering
Mangalam College of Engineering, Ettumanoor
Kottayam, India

Zacharia Alexander

Department of Computer Science and Engineering
Mangalam College of Engineering, Ettumanoor
Kottayam, India

Shilpa K Mohanan

Department of Computer Science and Engineering
Mangalam College of Engineering, Ettumanoor
Kottayam, India

Neena Joseph

Department of Computer Science and Engineering
Mangalam College of Engineering, Ettumanoor
Kottayam, India

Abstract—Traditional models of software development like waterfall model fails to cope up with the rapid changes in business, process and technology. Hence many software organizations are now moving from traditional process models to Agile methodologies for their projects. There are many frameworks under agile umbrella, popular frameworks being Scrum, eXtreme Programming, Crystal and Kanban. Scrum is the most popular framework followed in Indian companies. Many times organizations find it difficult to identify a suitable tool for managing their agile projects and many of them uses excel based templates.

Keywords—Agile, scrum, framework, sprint

I. INTRODUCTION

Agile helps to address to the needs of contemporary projects and is in sync with the current level of development. Continuous iterations and testing take place during the entire Software Development Life Cycle (SDLC) of a product, in Agile methodology. Scrum is the combination of Iterative model and Incremental model because of they are successive and incremental in terms of the features to develop an object oriented software.

Among many frameworks under agile umbrella, popular frameworks being Scrum, eXtreme Programming (XP), Crystal and Kanban, Scrum is the most popular framework followed in Indian companies. It is basically an agile, lightweight framework that provides steps to manage and control the software and product development process.

This paper aims to automate manual scrum boards used in organizations nowadays. Also, status of multiple projects (multiple Scrum boards) in an organization could be displayed on one device eg. say in four split windows or one project at a time on rotation. The dashboard shall also facilitate easy update of tasks status by the team members. It could be made available to software industry as an open source/free tool for managing their agile projects.

II. RELATED WORKS

Researches about the current status of Scrum, and its advancement in the last five years. The result reveals that Scrum is the popular software development methodology used by industries and it is also the interested area for the research community. It is capable to overcome the constraints of the traditional software development approaches in terms of providing simple and better implementations, better customer satisfaction and their involvement, faster delivery rate of the product, effectively versatile to new changing environment, and so on [1].

Software development methodologies are continually advancing due to changing technologies and new demands from clients. Agile methodologies are ideal for projects that show high inconsistency in tasks (because of changing requirements), in the capabilities of people, and in the technology being used. As Highsmith notes, they are also appropriate for projects where the value of the product to be delivered is vital to clients. Organizations must cautiously evaluate their status prior to stepping the path of agility [2].

The use of Agile Project Management (APM) practices at medium and large-sized Brazilian companies from various industry areas. Results show evidence of the presence of APM enablers, which create good conditions for implementing and adapting APM theory in non-software companies. The study also identified potential boundaries for APM implementation in traditional industries. There is a tendency to have less formalized processes, and therefore empowering the team with a level of autonomy to make decisions [3].

Principles of agile information systems development (ISD) have attracted both practice and research. The objective of this literature review is to approve, update and extend past reviews in terms of the general state of research on agile ISD. Besides including categories such as the employed research methods and data collection techniques, the importance of theory is

highlighted by assessing the theoretical foundations and contributions of former studies [4].

Sharing expert knowledge is one of the processes in developing software products. Since expert knowledge is generally tacit, the acquisition and sharing of tacit knowledge with the development of a Transactive Memory System (TMS) are significant factors in viable software teams. A theoretical model depicting the process for acquiring and sharing tacit knowledge and development of a TMS through social interaction presented [5].

Distributed software development is an important development approach for software companies as it brings enticing opportunities. There is little empirical knowledge on distributed agile software development. Attempts of research-based taxonomies for pinpointing the concept of agility, build the basis for a common definition, but most papers still rely on the Agile Manifesto that comprises of unverified principles and practices of practitioners which are not suitable as a strong theoretical grounding [6].

Agile methods have become the first choice for companies striving to improve their performance, but the methods were originally designed for small teams. Order is prioritized. Every item is called as User Story or Features. A longer sprint duration is preferred. Combining and customizing agile practices was also clear as many organizations viewed the agile method as continuously advancing [7].

III. PROPOSED METHODOLOGY

A. Development Methodology

Agile Scrum framework will be followed during development stage of the project. The project will be divided into four sprints of 2 weeks duration. User stories to be taken up in each sprint will be decided during sprint planning based on the priority of user stories and guidance from project guide. Task breakdown and estimation will be done during sprint planning in the beginning of every sprint.

Team will conduct daily scrum meeting to review the progress of project and completed functionality will be demonstrated to the project guide in every two weeks. There will be a retrospective meeting after the demo to reflect back on what went well and what went wrong in the current sprint, so that future sprints can be improved further.

B. Communication between Mobile and Web Server

In order to have communication between the mobile app and the web server, we use a PHP page which serves the requests coming from the app. The information is passed as parameters from the app. The PHP page also acts as an interface to the database that is, MySQL.

C. Graphical User Interface (GUI)

The users can access the system through GUIs provided in web and mobile applications. The mobile application will be developed with responsive GUIs compatible with android phones of 5" size and tablets of sizes 8" and 10".

IV. MODULES

A. Admin

□ The admin can register users by entering the Employee id, Employee Name and Email ID. The registered users can login to web and mobile application using employee id as user id and password received through email while registration .

Tasks:

- User Registration
- Create Project
- Edit Project
- Delete Project

B. Team Member

□ A team member can add tasks details such as task name, estimated hours, team member assigned for a selected user story in sprint backlog.

Tasks:

- Add Task
- View Task
- Edit Task
- Delete Task

C. Product Owner

The product owner can create a user story and add it to the product backlog of a project through web application

Tasks:

- Add User Story
- Edit User Story
- Delete User Story

V. IMPLEMENTING AGILE

Scrum framework allows you to implement Agile development methodology. Unlike the waterfall software development life cycle, the distinctive feature of Scrum is the iterative process of developing. Development divides into several phases. Each of them results into a ready-to-use product. At the end of each step (called sprint) a ready product is delivered to a customer. Customer's feedback helps reveal possible problems or change the initial plan, if needed.

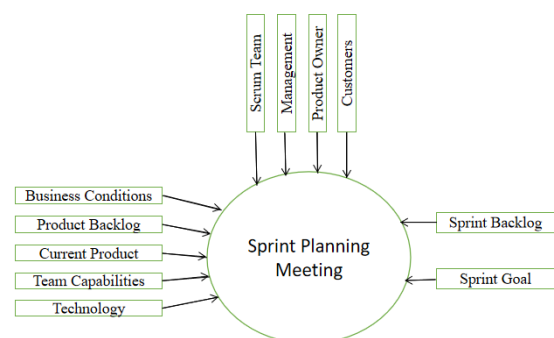


Fig. 1. Sprint Planning Meeting

Before we continue with Scrum, we should talk about the core roles involved into the process:

- Product owner- he takes care of the end user’s interests.
- Scrum master- should coordinate the whole process. His another task is to make sure that Scrum is used in a proper way. He also holds the Scrum meetings.
- Scrum team- develops the product. Its main tasks are programming, analysis, testing, etc.

VI. SYSTEM ARCHITECTURE

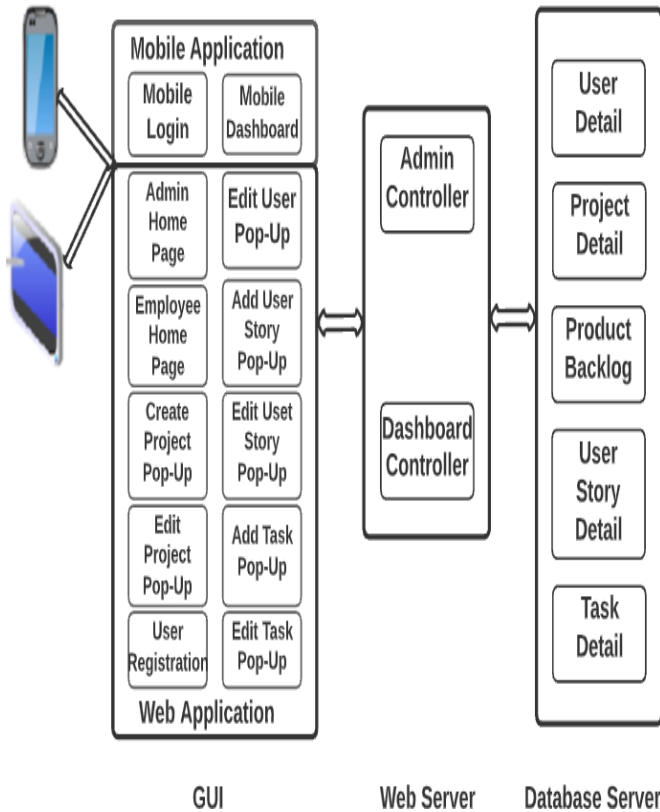


Fig. 2. System Architecture

The system consists of both web and mobile applications. These are provided with different interfaces. All the processing and computations are done in the web server. The details are stored on the database. System follows three-tier architecture with a Presentation/View tier, Controller tier and Data tier. View tier includes all Graphical User Interfaces (GUI) through which users interact with the system. Controller tier coordinates/controls View tier and Data tier for realizing various use cases. Data tier is responsible for managing persistent data (both read and write) in the system.

There are clearly defined interfaces between three tiers which makes system modular and makes system maintenance and enhancement easy. Modifications in one tier will not affect the other tiers as long as the interface doesn’t change. The mobile application will be developed with responsive GUIs compatible with android phones of 5” size and tablets of sizes 8” and 10”. A high level system architecture showing three tiers and various components in each tier is given in Fig.2.

VII. PHASES OF SCRUM MODEL



Fig. 3. Scrum Framework

A. Product Backlog Creation

Product backlog is a list that consists of features that should be implemented during the development process. It’s ordered by priority and its every item is called a User story. Every user story gets a unique ID. These are actual product requirements that were implemented during the XB Staff Manager Software developing process:

Steps 1 & 2: User Story Writing and Estimation

In agile world you start with writing user stories, sometimes in a special user story session. Since stories are written from the customer’s point of view so participation of customer and sponsor are required, at the same time representative of development team are also encouraged to be part of the workshop. Some of the user stories can be huge, and could be divided into many smaller user stories; those big user stories are also called “Epics” or “Features”. One Epic or Feature can have many smaller user stories.

Steps 3 & 4: Story Prioritization in the backlog

Only a set of stories are selected to be implemented in iteration. To select the stories for iteration, team prioritizes them based on their importance, need and risk value. For example in the start of the project user stories which are related to design will take precedence over development related user-stories. Usually higher risk stories are also of highest priority ones. Out of iteration stories can be prioritized and re-prioritize when needed inside the product backlog, but once added to the iteration (sprint) adding or replacing user-stories with the existing ones is not recommended.

There are couple of famous techniques used to prioritize the user stories. One of them is called MoSCoW (Must Should Could Won’t) technique. In this method user story’s priority is defined by words 'must', 'should', 'could' and 'won't'. Another way of prioritizing is to assign simple numbers to stories, story with lowest number has the highest priority, and story with highest number has lowest priority in the backlog.

Step5: Release Planning

A release is a time-boxed activity in which certain high ranked user stories are set to be completed. A release planning meeting is arranged so team could see what product owner and stakeholders want to achieve in a specific time. And also gets chance to look into the vision and road map for the product.

This meeting helps everyone in the team to come on the same page. Stakeholders and product owner transfer their vision and expectation to the development team. Development team contributes in refining the concept of the project by asking more questions and in some cases providing the answers of the unknowns given by product owner or come up as a result of the discussions.

B. Sprint Planning and Sprint Backlog Creation

Firstly, determine what your sprint’s duration will be. A short sprint allows you to release the working version of a product more frequently. As a result, customer’s feedback will be received more often and all the possible bugs and errors will be revealed in time.

As an alternative, you can prefer a longer sprint duration. It will allow developers to work more thoroughly. The optimal sprint duration is defined as an average of these two options. As a rule, a sprint lasts about 2 weeks. What’s more important at this phase is the cooperation between stakeholders and team members. The product owner determines the importance of a proper user story, while the scrum team defines the appropriate labor costs.

After that, the scrum team can select the most important user stories from the product backlog. Then team members should decide how they will solve this or that task. The Sprint backlog should be created next. It consists of user stories that will be completed during the current sprint. The amount of these stories depends on their duration in story points assigned to each story during evaluation stage. The team should be capable to finish all these stories on time.

Steps 6 & 7: Sprint Planning & Task Creation Process

The Sprint planning meeting takes place in the beginning of iteration. The development team selects a higher priority user story and divides it into one or many tasks. Tasks are created and assigned by developers.

Same process is repeated for all the stories until capacity of the team is reached. Once capacity is reached no more user stories are added. During the process of dividing the user stories into task, everyone in the development team collaborate and give his/her estimate to complete a task, planning poker technique can be used for that purpose. Team identifies the work to perform and commits to delivering potential product increment at the end of iteration.

Step 8: Sprint Iteration

Development team works on design, development and testing activities during an iteration (sprint). Duration of a sprint varies from team to team. 1-4 weeks iteration can be selected but mostly 2 week iteration is popular in agile.

Sprint iteration step shows sprint and activities performed during and at the end of a sprint. At the end of each sprint product output is produced and demonstrated to the customer in Sprint Review meeting. Also at the end of each sprint retrospective is performed to review the scrum process within scrum team. A release may consist of many sprints. Release output is a product that can be used by the customer.

C. Working on the Sprint. Scrum Meeting

After actual user stories for the current phase are chosen, the website development process begins.

To track the current working process, a task board is commonly used. There are usually big cards with the names of particular user stories and a bundle of little sticky notes with description of single tasks which are needed for implementation of this or that story. These cards are arranged according to their importance. When work on a task has been started, the corresponding sticker is moved from the “To do” field to the “In progress” one. When work is completed, sticker can be moved to the “Completed” field.

VIII. RESULT ANALYSIS

By comparing agile methodology with a traditional model, we can see some differences regarding their design and approach. Taking the case of agile model compared with waterfall model :

TABLE I. COMPARISON

	Agile Model	Waterfall Model
Design	Flexible	Structured
Testing	Concurrently with software development	Testing comes after the “Build” phase
Changes in project development requirement	Allows changes	No changes allowed once the project development starts
Product Delivery	Multiple small chunks of delivery at defined time intervals	One-time massive whole delivery
Planning Scale	Short-term	Long-term
Risk of project schedule	Low	High
Process		

From Table 1, it is clear that waterfall model is more stricter than agile model. And, using agile model will be more effective.

Considered a case of 30 candidates. Most of them had work experience more than 2 years. All are given projects and are being instructed to choose any methodology from the 3: Kanban, Scrum and Waterfall.

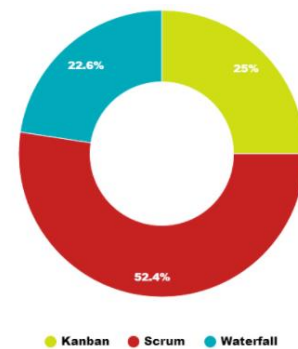


Fig. 4. Methodologies used by the candidates

Fig.4 shows the methodologies used by those 30 candidates. From the chart, it shows 52.4% of candidates used scrum, 25% used Kanban, and only 22.6% used Waterfall.

Most of the candidates participated in short term projects, which are completed in less than 6 months.

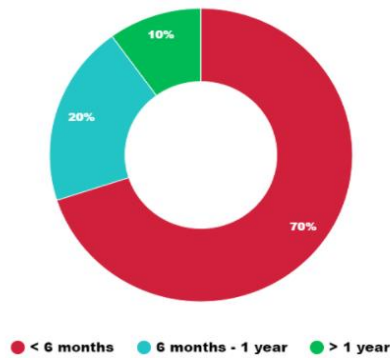


Fig. 5. Scrum project duration

Fig.5 shows the case of scrum projects. 70% of the projects are completed in less than 6 months, 20% within 1 year and 10% for ore than 1 year.

Thus, from the study it is clear that,

- Agile is the most flexible methodology, using Scrum.
- Candidates using Scrum had the most successful projects, even with the small term projects.

IX. CONCLUSIONS

Since the ideal result of every sprint is a working product, the testing process of full life-cycle is very important. There are different ways to minimize costs of the testing period. For example, you can decrease the overall amount of user stories. As a result, the number of possible bugs will be minimized.

We hope this project would revolutionize the existing scrum boards.

X. RISKS AND CHALLENGES

Following are the risks and challenges identified at this time:

- Availability of Smart/LED TV to test/demo the real usage of dashboard in a software organization.
- Creating real life project data for testing and demo (product backlog and sprint backlog of our own projects shall be captured in the system, in addition to data entry for 4 more dummy projects, in order to mitigate this risk).

REFERENCES

- [1] S. Sharma and N. Hasteer, "A comprehensive study on state of Scrum development," in Proc. Int. Conf. Comput. Commun. Autom., 2016, pp. 867–872
- [2] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large- scale agile transformations: A systematic literature review," J. Syst. Softw., vol. 119, pp. 87–108, 2016 EUROMICRO
- [3] E. C. Conforto, F. Salum, D. C. Amaral, S. L. da Silva, and L. F. M. de Almeida, "Can agile project management be adopted by industries other than software development?" Project Manage. J., vol. 45, no. 3, pp. 21–34, 2014
- [4] M. Hummel, "State-of-the-art: A systematic literature review on agile information systems development," in Proc. 47th Hawaii Int. Conf. Syst. Sci., 2014, pp. 4712–4721
- [5] S. Ryan and R. V. O'Connor, "Acquiring and sharing tacit knowledge in software development teams: An empirical study," Inf. Softw. Technol., vol. 55, no. 9, pp. 1614– 1624, 2013
- [6] M. Korkala and P. Abrahamsson, "Communication in distributed agile development: A case study," in Proc. 33rd Conf. Softw. Eng. Adv. Appl., 2007, pp. 203–210
- [7] S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of migrating to agile methodologies," Commun. ACM, vol. 48, no. 5, pp. 72–78, 2005