

## Overview on Estimation in Agile Methodology

Bhupinder Paul Singh Sahni\*

Bhupinder Paul Singh Sahni, USA

**Citation:** Sahni BPS. Overview on Estimation in Agile Methodology. *J Artif Intell Mach Learn & Data Sci* 2022, 1(1), 511-514.  
**DOI:** doi.org/10.51219/JAIMLD/bhupinder-paul/137

**Received:** 03 September, 2022; **Accepted:** 28 September, 2022; **Published:** 30 September, 2022

\*Corresponding author: Bhupinder Paul Singh Sahni, USA, E-mail: Bhupinder.sahni@gmail.com

**Copyright:** © 2022 Sahni BPS., Enhancing Supplier Relationships: Critical Factors in Procurement Supplier Selection.. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

### ABSTRACT

Software estimation gives the approximate calculation of software size, software development cost and effort, and development and testing schedule for a particular software project. Thus, Software estimation is a very important task in any software development methodology. Agile has been one of the commonly used Software Development Methodologies ever since its emergence in 2001. Since then, software companies have done many studies to investigate the best estimation techniques in agile. Traditional Effort Estimation techniques need well defined requirements, but Requirements tend to change in agile as Teams develop functionalities, based on customers inputs. So, Estimation in agile generally does not happen through traditional estimation techniques. In this paper we will study the basics of Agile methodology and different estimation techniques that software companies are commonly using to estimate Agile Projects.

**Keywords:** Estimation Technique; Agile methodology; Story Points; Planning Poker; Scrum; User Stories; Sprint

### 1. Introduction

Agile methodology of the software development is increasingly used across different domains. Agile Software development advocates adaptive planning, evolutionary development, early delivery, continual improvement, and flexible change in requirement at any time of the development process<sup>1</sup>. With adaptive nature and flexibility in changing requirements, Agile is the best suited for quick and effective software development. But with the flexibility that agile methodology provides based on the customer feedback, effort estimation for Agile Projects could be a very tricky and complex affair. There are many traditional Estimation techniques like Function point analysis, COCOMO, 3-Point Estimation, etc., but Agile uses more Subjective methods of estimation, which are more based on discussions within the team and coming to a consensus.

Like with every other Software Development methodology, Effort Estimation plays an important role in Agile software development as well. Effort estimation in Agile is the process of estimating the amount of work required to complete a task,

user story, or feature within a project. Agile teams typically use various techniques to estimate effort, with the goal of providing a prediction of how long it will take to complete the work. Agile is not about predicting exact timelines, but rather about understanding the overall workload and breaking it down into smaller portions of tasks and sub-tasks.

In this paper, we would study some of the basic terminologies used in agile methodology, like the Backlog, Agile, Sprint and User Stories. This would help us understand the objective of Agile methodology, how it works and in turn helps us decipher suitable estimation techniques that can be used in agile methodology.

### 2. Understanding Agile, Backlog, Sprint and User Stories

#### 2.1. Agile Methodology

Agile is a project management framework that gives a flexible approach to software development prioritizing adaptability, collaboration, and customer feedback. It focusses on iterative

and incremental development, where teams break down bigger requirements into smaller tasks, sub-tasks or stories, that helps team develop software incrementally<sup>2</sup>. Agile software development considers that production teams should start with simple and predictable approximations to the final requirement and then continue to increment the detail of these requirements throughout the life of the development<sup>3</sup>.

There are many variations in agile development for different types of projects. The most appropriate agile development model or variation depends on an organization’s structure and culture<sup>4</sup>. Some of the commonly used software development methods to support a broad range of Software Development Life Cycle (SDLC) includes<sup>5</sup>:

- Agile Unified Process (AUP)
- Extreme Programming (XP)
- Adaptive Software Development (ASD)
- Feature-Driven Development (FDD)
- Kanban
- Scrum
- Scaled Agile Framework (SAFe)

**Backlog:** In Scrum method of Agile, the basic concept for planning is the backlog, and there are 2 possible levels of backlog-Product Backlog and Sprint Backlog. Product Backlog considers all the requirements requested by the Customer or Business, broken down into stories. From the Product Backlog, the requirements or stories are considered to be developed which need to be delivered on priority based on customer needs and available time. This forms the Sprint Backlog<sup>6</sup>.

**Sprint:** In Scrum, projects progress via a series of iterations called sprints. The development team works to complete a set amount of work within the Sprint. Each sprint is typically 2-4 weeks long, with 2 weeks being the most common<sup>3</sup>. During a sprint, the team focuses on a specific set of tasks or user stories that have been selected from the product backlog. These tasks are prioritized based on their importance and value to the project. Sprints provides a structured approach to iterative development, allowing teams to regularly deliver working software and gather feedback from stakeholders.

**User Story:** A user story is a high-level definition of a requirement containing enough information so that the developers and testers can produce an estimate of the effort to implement it. Each story is formulated in one or two sentences in the language of the customer<sup>7</sup>.

### 3. Estimation Techniques for Agile Projects

A total of 25 primary studies were selected on agile estimation techniques. According to the study, Subjective estimation techniques (e.g. expert judgment, planning poker, use case points estimation method) are the most frequently applied in an agile context<sup>8</sup>. In this section, we will define some of the most commonly used Estimation techniques in Agile Projects. Agile estimation and planning methods are more effective than traditional methods, because agile methods concentrate on delivering value and establishing trust between the business and the project teams, keeping everything transparent, letting the business know of any changes that may come-up, challenges faced and thus adapting quickly to the changes after getting consent from Business<sup>7</sup>. Since Business is aware of each step

product team is developing or testing, they are quick to respond if there is anything that needs improvement.

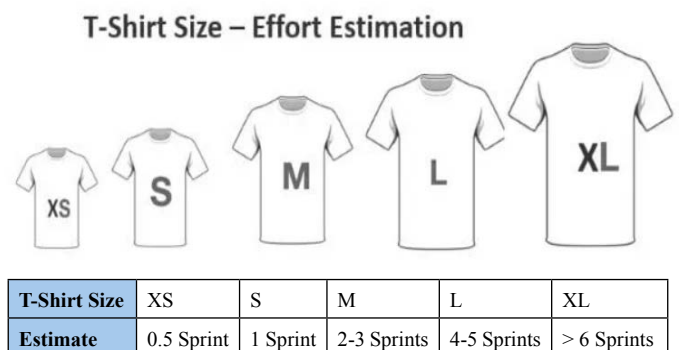
In an Agile project, it is common to begin iteration with incompletely specified requirements, the details of which will be discovered during the iteration<sup>9</sup>. The nature of agile may cause scope creeping due to changes in requirements which increases cost for software project<sup>10</sup>. A factor or extra time is added for the uncertainty that stakeholders think could be there and be discovered later. Requirements are first divided into stories, tasks and sub-tasks. Estimates are given on a story or task, and then relative estimates are provided for other stories, tasks or sub-tasks.

Most of the Agile based estimations are informal discussions on what the team thinks could be the estimate of a story, task or a sub-task. The estimates are given after the Product Owner defines a requirement; individual members of the team provide estimates using the estimation technique used by the Company. There could be a difference in estimates provided by one person to another, then the team have discussions on the estimates provided by each individual. These breaking down of requirements into stories, tasks and subtasks happen in Backlog Refinements and Sprint Planning Meetings. After the team discusses and reaches a consensus, Story points are allocated to stories, tasks or sub-tasks. The Business or the Customer then prioritizes these stories based on the initial estimates provided by the development team and on the business value of each story, task or sub-task<sup>11</sup>.

There are various methods for estimation used in agile software development, some of them are listed below:

#### 3.1. T-Shirt Size Estimation

T-Shirt Sizing is used at the initial phases of Agile projects to provide High level estimates of Tasks. High level estimates are provided according to T-Shirt Size of XS (Extra Small), S (Small), M (Medium), L (Large) and XL (Extra Large). It is an informal estimation technique and can be used to estimate quickly as it is relative sizing. The T-Shirt Size can then be further converted into number of Sprints<sup>12</sup>. Instead of giving a specific number, T-Shirt Size gives a range of estimation. Tasks which are first estimated using T-Shirt Size and given number of Sprints required to complete tasks, can then further be broken down into Sub-Tasks and estimated further using Planning Poker.



#### 3.2. Planning Poker

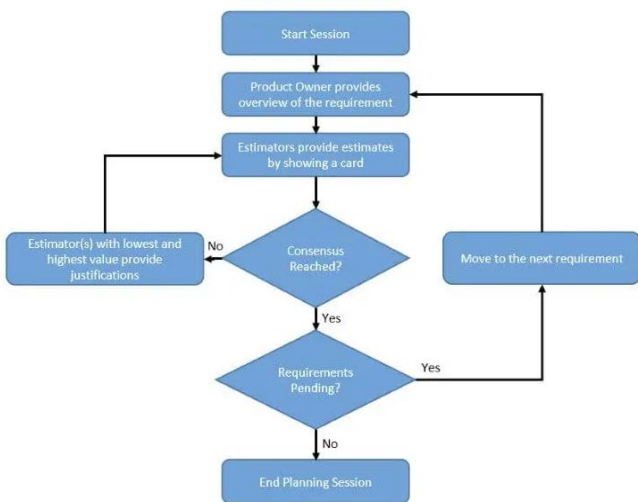
Similar to T-Shirt Size, Planning Poker is also a consensus-based estimation technique. In Planning Poker, estimation points usually follow Fibonacci sequence. The Fibonacci sequence is a series of numbers where each number is the sum of the two preceding numbers. The Fibonacci sequence is used in estimation because it represents a progressively increasing scale of complexity or effort required for tasks and stories or tasks are

estimated based on relative size<sup>13</sup>. The Fibonacci sequence used in Planning Poker has cards with estimation cards as 1, 2, 3, 5, 8, 13, 21, 34. If a story or task is estimated to be more than 8 points, then it is broken down into smaller stories, tasks or sub-tasks.



If all estimators selected the same value, then that value becomes the estimate of that story, task or sub-task. If the estimates do not match, then the estimators discuss the reasons behind choosing their estimates and have a discussion. After discussion, estimators can re-elect their estimates again till it comes to a consensus<sup>14</sup>.

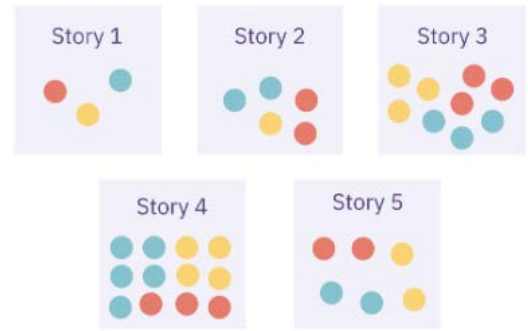
Planning Poker Flowchart



### 3.3. Dot Voting

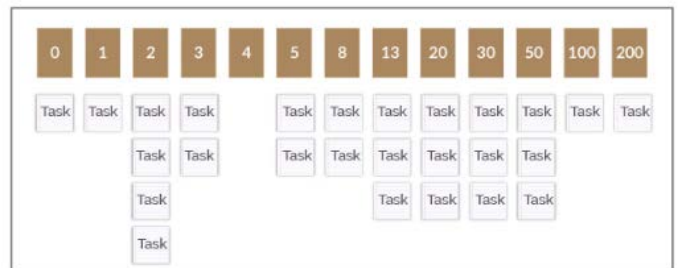
Dot voting, also known as dot democracy, is a simple and effective technique used to prioritize items or make decisions in a group setting. Dot Voting method is the ranking between highest priority story to lowest priority stories from Product Backlog<sup>5</sup>. In this method each team member gets a small number of dots in the form of stickers and uses them as votes to indicate the size and importance of an item. The more votes an item gets, the bigger its size, and the higher its priority. Dot voting encourages active participation from all members, prevents dominance by a few individuals, and provides a visual representation of the group’s collective priorities.

### Dot voting



### 3.4. Bucket System

The main purpose of this technique is to briefly estimate a large backlog and assign a bucket. In a collaborative approach, stories, tasks and sub-tasks are categorized into different buckets or groups, based on complexity, different criteria and attributes<sup>5</sup>. For example, items with high business value and low effort might be placed in a “High Priority” bucket, while items with low business value and high effort might be placed in a “Low Priority” bucket. Buckets can also be in the form of Fibonacci sequence or a different sequence of numbers. The bucket system is flexible and adaptable, making it suitable for a wide range of Agile projects and teams. It provides a transparent and visual way to prioritize work, fostering collaboration and alignment among team members.



### 4. Conclusion

Customer change requests and unclear requirements are the main causes of the failure of agile software estimation, and Agile Software development focusses on evolving requirements as the functionality is developed and the customer provides feedback. So, effort estimation in agile plays a crafty and critical role in ensuring project is delivered within the timelines. Subjective estimation techniques, e.g. expert judgment-based techniques, planning poker or T-Shirt Size, are the ones used the most in agile effort estimation. But it is important for Agile teams to regularly review and refine their estimation techniques based on past performance and feedback to improve accuracy over time.

### 5. References

1. Shimoda A, Yaguchi K. A method of setting the order of user story development of an agile-waterfall hybrid method by focusing on common objects. 6th IIAI International Congress on Advanced Applied Informatics (IIAIAI) 2017; 301-306.
2. Begel A, Nagappan N. Usage and perceptions of Agile software development in an industrial context:an exploratory study. First International symposium on ESEM 2007; 255-264.

3. Kumar G, Bhatia P. Impact of Agile methodology on software development process. *IJCTEE* 2012;2: 46-50.
4. Harb Y, Noteboom C, Sarnikar S. Evaluating project characteristics for selecting the best-fit agile software development methodology: A teaching case. *JMWAIIS* 2015;1: 34-47.
5. Mallidi R, Sharma M. Study on Agile story point estimation techniques and challenges. *IJCA* 2021;174: 9-14.
6. Buglione L, Abran A. Improving Estimations in Agile Projects: Issues and Avenues. *Software Measurement European Forum* 2007; 265-274.
7. Cohn M. *Agile Estimating and Planning*. Addison-Wesley 2005.
8. Usman M, Mendes E, Weidt F, Britto R. Effort estimation in agile software development: A systematic literature review. *Proceedings of the 10th International Conference on Predictive Models in Software Engineering* 2014; 82-91.
9. Osman H, Musa M. A survey of agile software estimation methods. *IJCST* 2016;7: 38-42.
10. Keaveney S, Conboy K. Cost estimation in agile development projects. *ECIS 2006 Proceedings*. 2006; 169.
11. Lovaasen G. Brokering with eXtreme Programming. In *XP Universe* 2001.
12. Seth A. Agile estimation using T-Shirt Sizing. *PM Tips* 2020.
13. Tanmrakar R, Jorgensen M. Does the use of Fibonacci numbers in planning poker affect effort estimates. *Proceedings of EASE* 2012; 228-232.
14. Seth A. Effort Estimation using Planning Poker. *PM tips* 2020.