**Story Points**

Groomed, Ready User Stories have high-level estimates. These estimates would be used by the Product Owner to create a list of prioritized approved User Stories which have been estimated by the Scrum Team. Such estimated User Stories are then committed to be completed by the Scrum Team during the Sprint.[[1]](#footnote-1)

**Estimation Criteria**

The primary objective of using Estimation Criteria is to maintain relative estimation sizes and minimize the need for re-estimation. Estimation Criteria can be expressed in numerous ways, with two common examples being story points and ideal time. For example, an ideal time normally describes the number of hours a Scrum Team member works exclusively on developing the project’s deliverables, without including any time spent on other activities or work that is outside the project. Estimation Criteria make it easier for the Scrum Team to estimate effort and enable them to evaluate and address inefficiencies when necessary.[[2]](#footnote-2)

**User Story Estimation Methods**

Tools used for *Approve, Estimate, and Commit* User Stories can be used for creating high level estimates for Epic(s) when we create the Prioritized Product Backlog. [[3]](#footnote-3)

Some important tools are[[4]](#footnote-4):

* Planning Poker
* Fist of Five
* Points for Cost Estimation
* Other Estimation Techniques

**Planning Poker**

Planning Poker, also called Estimation Poker, is an estimation technique which uses consensus to estimate relative sizes of User Stories or the effort required to create them.[[5]](#footnote-5)

In Planning Poker, each team member is assigned a deck of cards. Each card is numbered in a sequence and the numbers represent complexity of the problem, in terms of time or effort, as estimated by the team member. The Product Owner chooses a User Story from the Prioritized Product Backlog and presents it to the team. The Scrum Team members assess the User Story and try to understand it better before providing their estimate for developing it. Then, each member picks a card from the deck that represents their estimate for the User Story. If the majority or all team members select the same card then the estimate indicated by that card will be the estimate for that User Story. If there is no consensus, then the team members discuss reasons for selecting different cards or estimates. After this discussion they pick cards again. This sequence continues until all the assumptions are understood, misunderstandings are resolved, and consensus or agreement is reached.[[6]](#footnote-6)

Planning Poker advocates greater interaction and enhanced communication among the participants. It facilitates independent thinking by participants, thus avoiding the phenomenon of group think.[[7]](#footnote-7)

*Note: Planning poker is a fun effective way to ensure that all team members have input into the estimating the teams’ work. To avoid group think, and to ensure that all members get a open say in the analysis, cards should be placed face down and all turned over at the same or as I like to do, on the count of 3 everyone shows their cards*.

**Fist of Five**

Fist of Five is a simple and fast mechanism to achieve consensus in a group and drive discussion. After initial discussion on a given proposal or a pending decision, the Scrum Team members are each asked to vote on a scale of 1 to 5 using their fingers. The value in using this technique is not only consensus building but also driving discussion because each team member is asked to explain the reason for their ranking. They are also given the opportunity to express any issues or concerns. Once the team has discussed it, a collective decision will be made.[[8]](#footnote-8)

The number of fingers used to vote indicates the level of agreement and desire for discussion[[9]](#footnote-9):

* One finger: I disagree with the group's conclusion and have major concerns.
* Two fingers: I disagree with the group's conclusion and would like to discuss some minor issues.
* Three fingers: I am not sure and would like to go with the group's consensus conclusion.
* Four fingers: I agree with the group's conclusion and would like to discuss some minor issues.
* Five fingers: I wholeheartedly agree with the group's conclusion.

**Points for Cost Estimation**

Cost estimation can be accomplished through the use of relative units (e.g., effort estimates) rather than absolute units (i.e., actual costs incurred). In order to estimate the cost to implement a User Story, the Scrum Team can use story points. When this is done, the cost estimated for each task will be in the form of story points, rather than monetary units. In order to do this successfully, the Scrum Team should identify a baseline User Story that all team members can relate to. Once this baseline is identified, all cost estimates for User Stories should be done compared to that baseline. These estimates remain fixed throughout a Sprint because teams are not supposed to change during a Sprint. [[10]](#footnote-10)

**Other Estimation Techniques**

**Wideband Delphi**

Wideband Delphi is a group-based estimation technique for determining how much work is involved and how long it will take to complete. Individuals within a team anonymously provide estimations for each feature and the initial estimates are plotted on a chart. The team then discusses the factors that influenced their estimates and proceed to a second round of estimation. This process is repeated until the estimates of individuals are close to each other and a consensus for the final estimate can be reached.[[11]](#footnote-11)

Planning poker is one example of a Wideband Delphi technique. It is also important to note that it is the individual input collected by a mechanism that avoids the group thinking. Then the individual inputs are used for a group decision.[[12]](#footnote-12)

**Relative Sizing/Story Points**

In addition to being used for estimating cost, story points can also be used for estimating the overall size of a User Story or feature. This approach assigns a story point value based on an overall assessment of the size of a User Story with consideration given to risk, amount of effort required, and level of complexity. This assessment will be conducted by the Scrum Team and a story point value will be assigned. Once an evaluation is done on one User Story in the Prioritized Product Backlog, the Scrum Team can then evaluate other User Stories relative to that first story. For example, a feature with a 2 -point story value must be twice as difficult to complete as a feature with a 1-point story; a 3-point story should be three times as difficult to complete as a 1-point story.[[13]](#footnote-13)

**Affinity Estimation**

Affinity Estimation is a technique used to quickly estimate a large number of User Stories. Using sticky notes or index cards and tape, the team places User Stories on a wall or other surface, in order from small to large. For this, each team member begins with a subset of User Stories from the overall Prioritized Product Backlog to place by relative size. This initial placement is done in silence. Once everyone has placed their User Stories on the wall, the team reviews all of the placements and may move User Stories around as appropriate. This second part of the exercise involves discussion. Finally, the Product Owner will indicate some sizing categories on the wall. These categories can be small, medium, or large, or they may be numbered using story point values to indicate relative size. The team will then move User Stories into these categories as the final step in the process. Some key benefits of this approach are that the process is very transparent, visible to everyone, and is easy to conduct.[[14]](#footnote-14)

**Estimate Range**

Estimates for projects should be presented in ranges. Precise figures may give an impression of being highly accurate when in fact they may not be. In fact, estimates by definition are understood not to be precisely accurate. Estimate ranges should be based on the level of confidence the team has in each estimate. The range can be narrow when the team is confident and wide when the team is less confident[[15]](#footnote-15)

**Triangulation[[16]](#footnote-16)**

Analogy: “This story is a little bigger than that story.” When estimating by analogy, the estimator compares the story being estimated with one or more other stories. If the story is twice the size, it is given an estimate twice as large. There is evidence that we are better at estimating relative size than we are at estimating absolute size (Lederer and Prasad 1998; Vicinanza et al. 1991).[[17]](#footnote-17)

When estimating this way, you do not compare all stories against a single baseline or universal reference. Instead, you want to estimate each new story against an assortment of those that have already been estimated. This is referred to as triangulation. To triangulate, compare the story being estimated against a couple of other stories. To decide if a story should be estimated at five story points, see if it seems a little bigger than a story you estimated at three and a little smaller than a story you estimated at eight.[[18]](#footnote-18)

**Self-Assessment**

**Estimating**

Understand the different estimation levels in Scrum[[19]](#footnote-19):

* That product backlog items and sprint backlog items (including tasks) are each estimated.
* One or more estimating units for product backlog items, such as story points or ideal days.

Understand that the purpose of estimating[[20]](#footnote-20):

* That the product backlog estimates enable the product owner to prioritize work and the Scrum team to determine the effort remaining to deliver the release thereby tracking the progress and creating a release plan.
* That the task estimates help the team to pull the right number of product backlog items into the sprint allowing the team to make a realistic commitment.

Understand that the accuracy of an estimate is more important than the precision of the estimate[[21]](#footnote-21):

* The difference between accuracy and precision
* That an accurate estimate may be useful even if its precision is less than the product owner would like.
* That an inaccurate (but precise) estimate is of no use.
* That estimates in Scrum are team estimates and should not include any assumptions about who is going to implement a product backlog items or sign up for a task.

Understand that estimates of size and duration can be done separately[[22]](#footnote-22)

* That the team should estimate the size of the effort and then velocity should be used to empirically determine the duration. Size and duration can vary independently.

Understand the impact of pressuring team members to provide low estimates[[23]](#footnote-23)

* That team members will provide low estimates when forced but that they won’t finish the work within that time.
* That excessive time pressure leads to cutting quality (which comes back to hurt the project later), low morale, and loss of creativity.

Understand the difference between estimating and committing[[24]](#footnote-24)

* That an estimate is a prediction of the future and always includes some amount of uncertainty, even if that uncertainty isn’t stated
* That a commitment is based on an estimate, which must be created first
* The implications of equating estimating with committing

1. Scrum Book of Knowledge, [www.scrumstudy.com](http://www.scrumstudy.com) [↑](#footnote-ref-1)
2. SBOK [↑](#footnote-ref-2)
3. SBOK [↑](#footnote-ref-3)
4. SBOK [↑](#footnote-ref-4)
5. SBOK [↑](#footnote-ref-5)
6. SBOK [↑](#footnote-ref-6)
7. SBOK [↑](#footnote-ref-7)
8. SBOK [↑](#footnote-ref-8)
9. SBOK [↑](#footnote-ref-9)
10. SBOK [↑](#footnote-ref-10)
11. SBOK [↑](#footnote-ref-11)
12. SBOK [↑](#footnote-ref-12)
13. SBOK [↑](#footnote-ref-13)
14. SBOK [↑](#footnote-ref-14)
15. BOK [↑](#footnote-ref-15)
16. <https://www.mountaingoatsoftware.com/system/asset/file/15/aep_sample.pdf> [↑](#footnote-ref-16)
17. MGSW [↑](#footnote-ref-17)
18. MGSW [↑](#footnote-ref-18)
19. CSPO Learning Objective, [www.scrumalliance.org](http://www.scrumalliance.org) [↑](#footnote-ref-19)
20. CSPO [↑](#footnote-ref-20)
21. CSPO [↑](#footnote-ref-21)
22. CSPO [↑](#footnote-ref-22)
23. CSPO [↑](#footnote-ref-23)
24. CSPO [↑](#footnote-ref-24)