Story Point Estimating

by Richard "Dick" Carlson

ALEA

Agile & Lean Education Associates

"Teaching Agile & Lean from A to Z"

Story Point Estimating

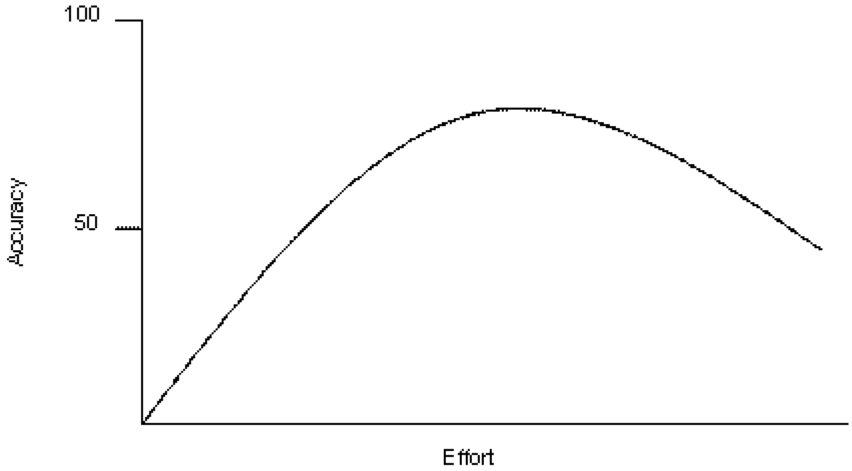
Dick Carlson, Agile & Lean Education Associates (ALEA)

Abstract

Traditional and Agile projects measure size differently. Agile software project estimates are based on story points or ideal days. Story points are a unit of measure for expressing the overall size of a user story, feature, or other piece of work. Ideal day estimating on a software project is easier to estimate at first, but as the natural overhead present in projects every day is experienced, working on planned activities becomes difficult.

Accuracy of Estimates

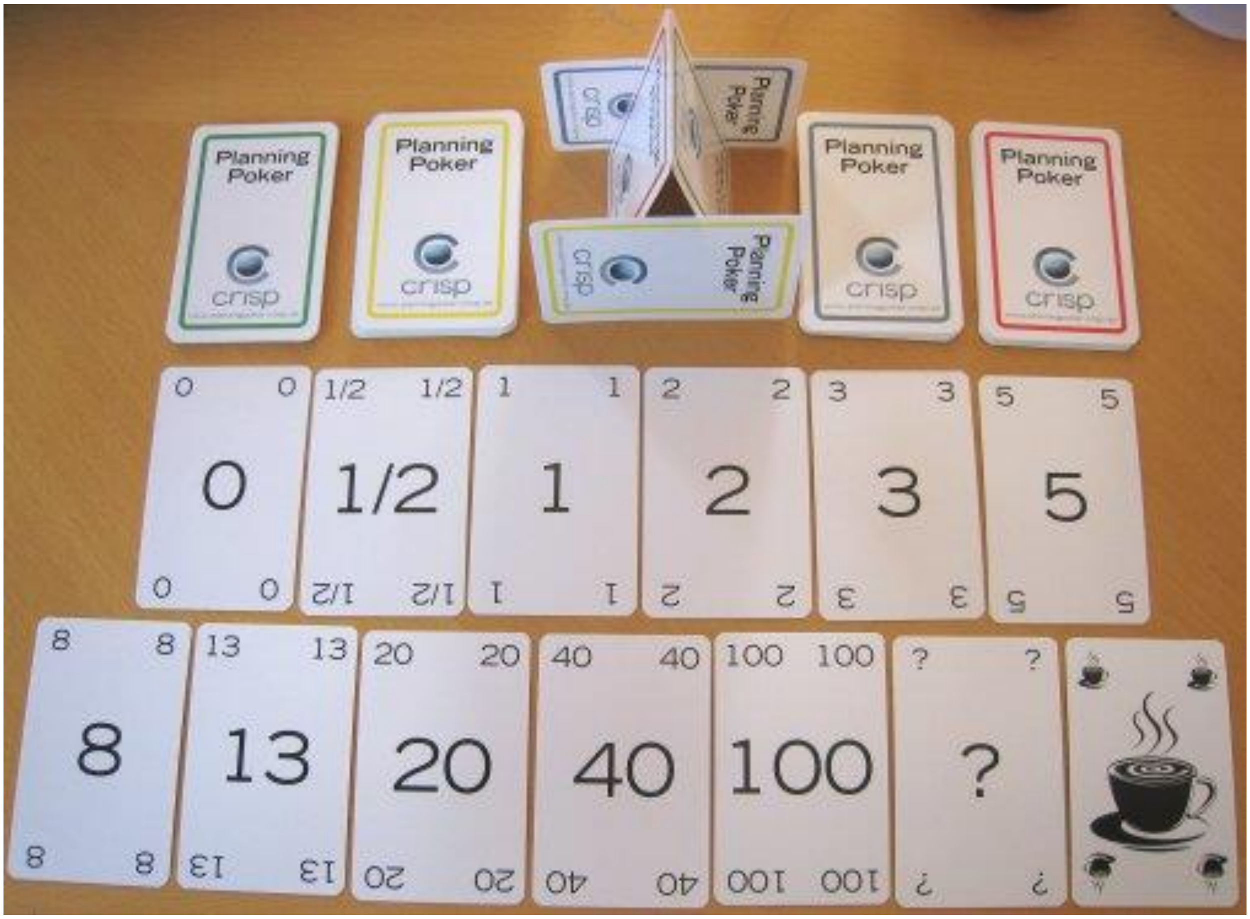
The most important thing to remember about estimates is, "No matter how much effort is put into an estimate, an estimate is still an estimate." No amount of additional effort will make an estimate perfect or finite. In the figure below, note that no matter how much effort is invested, the estimate is never at the top of the accuracy axis. Also note how little effort is required to move the accuracy up dramatically from the baseline. About 10% of the effort gets 50% of the potential accuracy. Also note that eventually the accuracy of the estimate declines. Too much estimating effort typically results in a less accurate estimate.



Story Point Estimating

Story points help drive cross-functional behavior and do not decay. When estimating with story points, a point value is assigned to each item. The raw values that are assigned are unimportant. What matters most are the *relative values*. That is, a story valued at 2 story points should be twice the size of a story valued at 1 story point. For example, if assigning points to the size of dogs, we might assign 8 doggie points to a Great Dane and 5 doggie points to a German shepherd.

The number of story points associated with a story represents the overall size of the story. There is no set formula for defining the size of a story. A story point estimate is the "bigness" of work that is influenced by how difficult it seems or is, and how much of it there is to build. Depending on who you talk to, who you are trained by, or whose book you read, story point ranges will vary. Some use a range of 1–100. Some Agile experts claim that a range of 1–10 provides optimum accuracy. Units based on the *Fibonacci* sequence seem to work best (e.g. 1, 2, 3, 5, 8, 13, etc.). (Note: A number in the Fibonacci sequence is created by taking the sum of the previous two numbers.) I prefer to use the more common Agile estimating technique, *Planning Poker*, with new teams (see illustration below). As the team becomes accustomed to using Planning Poker, you should observe a more efficient departure from "card playing" to verbal or visual numerical expressions among some team members. If you are the team's coach or Scrum Master, I would recommend that you follow through with this trend and engage the entire team in transitioning to the more efficient estimating method. Doing so should result in a 30-40 percent savings in estimating activity.



Source: http://www.crisp.se/bocker-och-produkter/planning-poker

Ideal Day Estimating

An NFL football game is divided into four 15-minutes quarters, but the time spent from when the teams enter the field to the end of the game may take two or three hours, which is elapse time. Ideal time is used by the officials to keep track of time allotted in each quarter. Planned activities are affected when people spend time to read and answer email, make phone calls, attend meetings, address personal issues, attend training, support releases, management reviews, take sick time, enjoy vacations, perform task switching, assist in fixing software bugs, attend buddy/peer reviews, and participate in other activities that interfere with meeting commitments.

Problems become more apparent when a team member is asked, “How long will this take?” The typical response might be, “In five days.” So management counts off five days on its calendar and marks the day with a big red X. It’s likely the team member really meant to say, “Five days if that’s all I do, but if I have to do a lot of other things, probably ten days.”

Multi-tasking broadens the gap between ideal time and elapse time. If you choose to estimate in ideal days on your software project, you may assume:

- The story being estimated is the only thing you will work on,
- Everything you will need will be available to you when you start, and
- There will be no interruptions.

If we estimate the number of ideal days a user story will take to complete, factoring in impacts from environmental overhead or bureaucracy will not be necessary, because the actual time needed to complete the story will still take one ideal day. Of course, the amount of elapse time will very significantly. For example, if I estimated creating a login screen will take me one ideal day, the amount of elapsed time may be five days.

Another thing about ideal time is that it cannot be established as a standard for doing work, because “My ideal time is not the same as your ideal time.” For example, something that I estimate as one ideal day of work may be estimated by another person as two ideal days to complete.

Sprint Planning

Scrum teams often use Planning Poker for estimating product backlog work that combines expert opinion, analogy, and disaggregation for quick, reliable estimates. Only Team members may participants in Planning Poker. While expert opinions are appreciated and welcomed, they cannot be used by a Scrum team, because unless the expert is going to do the actual work, the estimate should not be applied. The Product Owner should participate in Planning Poker, but does not participate in estimating.

(Note: In my experience, I have found that noteworthy (i.e. active and participating) Product Owners add tremendous value to the estimating process by responding to questions the team may have regarding work that must be estimated. Also, Product Owners who serve as part-time team members must participate actively in estimating.)

- **Expert opinion:** In an expert, opinion-based approach to estimating, an expert is asked how long something will take or how big it will be. The expert relies on intuition or gut feel and provides an estimate.
- **Analogy:** 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’s given an estimate twice as large.
- **Disaggregation:** Disaggregation refers to splitting a story or feature into smaller, easier-to-estimate chunks.

At the start of sprint planning, each team member is given a deck of cards. Each card contains a valid estimating number 1, 2, 3, 5, 8, 13, 20, 40, and 100, where 1 is the smallest, and 100 the largest. For each user story to be estimated, a moderator (preferably the Product Owner) reads the description. The goal of Planning Poker is to derive an estimate that is valuable and where an economical estimate can be made. After all questions are answered, each team member selects a card with a number that represents the member's estimate. Cards are not shown until each team member has made a selection, and at that time, all cards are turned over simultaneously so that everyone can see all of the estimates.

For distributed teams, there are other ways estimating can be played. On one of my virtual sprints, where using Planning Poker cards were not practical, we used Instant Messaging (IM) with all team members, the Product Owner, and the Scrum Master who facilitated the estimating session. The Planning Poker method (numbering) was used. The Scrum Master facilitated the estimating game, who asked team members to select an estimation number, and then on his signal, had everyone send their estimate simultaneously via IM. Planning Poker estimating method variations are recommended and encouraged to make the estimating process more efficient and effective.

Estimates will likely differ significantly, and the outlier estimates should be explained so the team understands why the numbers selected were used. The team then can discuss the story in more detail and adjust its estimates accordingly, or until the team converges on a single estimate. The goal is for the team to converge on a single estimate that can be applied to the story. It isn't necessary for everyone to turn over cards with the same estimate. If on the second round with five estimators showing estimates of 5, 5, 5, 5, and 3, then ask the low estimator if he or she is okay with an estimate of 5. Conversely, if on the second round with five estimators showing estimates of 5, 5, 5, 5, and 7, then ask the low estimator if he or she is okay with an estimate of 5. The point is not absolute precision, but everyone should be in agreement and the estimate should be reasonable.

Scrum teams typically estimate two different times. During the first round, there usually is a need to estimate a large number of product backlog items before the project officially starts (during Sprint 0) or on the first day of the sprint. Estimating an initial set of user stories may take two or three meetings lasting from one to three hours, depending on the number of user stories, the size of the team, and the Product Owner's ability to clarify the requirements succinctly. Teams should estimate any new stories identified during a sprint. This can be done by holding a short estimation meeting towards the end of a sprint, and is usually sufficient for estimating any work that is introduced during the sprint that allows new work to be considered in the prioritization of the next sprint.

The Planning Poker estimating technique works for:

- Team members who will estimate and do the work
- Team members who are required to justify their estimates
- Focusing most estimates within an approximate single increment or decrement
- Combining individual estimates through group discussion to provide better estimates
- Constraining estimates to a set of values so time is not wasted in meaningless arguments

In summary, using the Planning Poker estimating technique ensures that everyone's opinion is heard, reduces or eliminates wasteful time, and adds an element of fun into the estimating process. What can be better than that?

Velocity

Velocity is a measure of a team's rate of progress, and it is calculated by summing the number of completed story points assigned to each user story that is completed during a sprint. So, if a team completes three user stories estimated at 5 points each, the velocity is 15 story points. If the team completes two 5-point stories, the velocity is 10 story points. If a team completed 10 story points of work during the last sprint, it has a good chance of completing 10 story points of work during the next sprint provided the team's membership does not change; there are no significant impediments, and the sprint's duration remains the same. Because story points are estimates of relative size, this will apply whether the team works on two 5-point stories or five 2-point stories.

The amount of story points completed by one team for the same work may not be the same for another team. If exactly the same function were implemented by two different teams, it would not be surprising if each team assigned a different story-point value. This may sound silly, but consider that many teams are using Planning Poker cards as their primary tool for estimating. Team size is a factor in determining velocity. I have seen 10-12 man teams complete more than 200 story points in two-week sprints after they had stabilized.

Summary

Story points are a relative measure of the size of a user story. A user story estimated as a 10-point story is twice as big, twice as complex, or twice as risky as a story estimated as a 5-point story. A 10-point story is half as big, half as complex, or half as risky as a 20-point story. Determining the size of a user story is estimated using the Planning Poker estimating technique.

Story points are purely an estimate of the size of the work to be performed. The duration of a project is not estimated as much as it is derived by taking the total number of story points and dividing it by the team's velocity. Always remember that no matter how much effort is put into an estimate, an estimate is still an estimate.

Velocity is a measure of a team's rate of progress per iteration. At the end of each sprint, a team can include all stories it has completed to calculate the velocity by summing the story point estimates for each completed story.

References

User Stories Applied, Mike Cohn, 2004, Addison-Wesley

Agile Estimating and Planning, Mike Cohn, 2006, Prentice Hall

About the Author

Dick Carlson has an extensive engineering background that includes many years of practical knowledge and hands-on experience in the implementation and deployment of Agile, Lean, and Scrum values and principles in communications-electronics engineering, software engineering, and systems engineering within the aerospace, DoD, IT, and industry domains. He has developed and actively conducted comprehensive training courses for Scrum Teams, Scrum Masters, Product Owners, project/program managers, customers, executives, organizational leaders, and others interested in learning how to implement and deploy Agile, Lean, and Scrum.

Recently retired from Boeing, Dick has been an active transformational leader for many small and large Agile projects, and frequently shares his experiences of successful Agile implementation at conferences, workshops and symposia. He regularly counsels executives and leaders on the benefits of using Agile, actively coach's teams on Scrum and Lean-Agile Project

Management fundamentals, and then follows up with mentoring activities to ensure successful project implementation.

Mr. Carlson has a Bachelor of Science degree from the University of Maryland, and is a Certified Scrum Professional, Certified Scrum Master, and Certified Scrum Product Owner, and holds a Lean-Agile Project Management certification. Mr. Carlson has presented Agile topics at Software Technology Conferences every year since 2010.