Transcript

Estimating User Story Size

Learning Objectives

After completing this topic, you should be able to

- *recognize activities carried out when using the planning poker technique*
- *describe the ideal days estimation technique*

1. Estimating using story points

Once you've gathered project requirements in the form of user stories, the final step in determining scope and finalizing the release plan involves using a rough estimate of how much work a team can develop in an iteration – their velocity – to figure out how many backlog items or user stories the team will be able to finish prior to the release date. This is where estimating becomes a critical factor in project planning.

The focus when initially planning an agile project is on estimating outcomes, rather than the activities or tasks required to achieve them. This is in keeping with the agile emphasis on what's of value to the customer.

When estimating for outcomes, you should come up with the best possible estimates for the user stories that will be developed during the project. It's particularly important to have good estimates for the most important backlog items. This includes user stories that are more likely to be selected for inclusion in an upcoming iteration or release milestone, such as new features that the product owner wishes to include most.

The team should come up with estimates for all backlog items identified by the customer but may simply assign large story points to user stories that aren't as well understood and will likely be refined further and developed during a later iteration or release.

In an agile approach, estimation should be a group – rather than individual – activity. It should also involve people who have expertise on the type of product you're creating. Generally all the members of a project team collaborate to create the required estimates.

Estimates are typically more reliable when they're derived collaboratively for two main reasons:

- anyone can potentially be assigned any task, so the estimate should be based on input from everyone in the development team – there's no guarantee the person making an estimate will be the person to perform a particular task, and

- everyone in a group can contribute different experiences and expertise, with the result that estimates are more accurate

If you're used to a more traditional approach to project management, you may have misconceptions about what creating estimates for an agile project involves. You may think it involves estimating what each user story will cost, or how much time each story will take to complete.

Select each misconception for more information about it.

Estimating cost

In an agile approach, you don't estimate how much money you'll spend on developing each user story. Instead developing estimates during project or release planning is meant to create a shared understanding of the relative size and complexity of each user story.

Estimating time

In a traditional approach, the focus is on estimating how long it will take to perform the required project work, or the date when the work will be completed.

In an agile approach, the focus is on estimating the development effort that's required. It's accepted that initial attempts to predict story durations are likely to be inaccurate, and that the time it takes to complete tasks will depend on the capacity of the development team.

When estimating the relative sizes of user stories, agile teams often use *story points*. Each story point represents a fixed amount of development effort, or work.

For example, a fairly small coding task might be assigned five story points. If another task is judged to require three times as much work, it will be assigned 15 story points. On their own, story points don't tell you how long each user story will take to develop. Story points should not be seen as equivalent to hours and the actual story point value assigned could vary greatly from one team to the next. The key is to use them as ratings of size in relation to other stories in the backlog.

It's important not to confuse the size of a task – which is measured in story points – with its complexity. Although size can sometimes be an indicator of complexity, it's possible that a relatively simple story will involve as much work, and therefore be allocated the same number of points, as a complex one.

For example, the task of capturing data in order to populate a database is simple but, depending on the amount of data that's involved, might involve as much or more work than a far more complicated task, like designing a system's user interface.

Question

What do story points represent?

Options:

1. The relative difficulty of developing each user story
2. The number of hours it will take to complete a user story
3. The relative size of each user story in terms of the work effort required to develop it
4. The relative cost of developing each user story

Answer

Option 1: *Incorrect. Story points don't indicate the difficulty, or complexity, of developing particular user stories. Instead they represent the relative amount of development effort involved in developing user stories.*

Option 2: *Incorrect. Story points don't indicate how long it will take to develop user stories. They are used to allow the team to size user stories relative to other stories in the backlog.*

Option 3: *Correct. An agile team uses story points to express estimates of the relative sizes of backlog items or user stories, in terms of how much development effort it will take to develop them.*

Option 4: *Incorrect. Story points don't provide an indication of what it will cost to develop user stories.*

Correct answer(s):

3. The relative size of each user story in terms of the work effort required to develop it

Question

Your team is using story points to help plan a product release date. So far the team has assigned 1 point to story A, 5 points to story B, and 2 points to story C.

Which observations about these user stories are correct?

Options:

1. Story B is expected to take more than twice as long as story C to complete
2. Story C is expected to involve twice as much effort as story A

3. Story B is expected to involve five times the work involved in story A
4. Story C is twice as complex as story A

Answer

Option 1: *Incorrect. On their own, story points don't represent how long user stories, or particular development tasks, will take to complete. Instead they represent the amount of effort each user story will take to develop, relative to the other stories in a project.*

Option 2: *Correct. User story C has been assigned twice the number of story points as user story A. This means that, according to the team's estimates, developing story C is likely to involve twice the amount of development effort as developing story A.*

Option 3: *Correct. User story B has been assigned five times the number of points assigned to user story A. This indicates that story B will involve five times the amount of work, or development effort, as story A.*

Option 4: *Incorrect. It's inaccurate to use story points as a gauge of a user story's complexity. A story that's straightforward to develop may involve the same amount of work as one that's far more complex.*

Correct answer(s):

2. Story C is expected to involve twice as much effort as story A
3. Story B is expected to involve five times the work involved in story A

2. Using planning poker

A popular agile estimation technique that uses story points is known as *planning poker*. Planning poker involves using a pack of agile planning cards, arranged in a deck, to represent possible numbers of story points for assigning to user stories. It's an adaptation of a technique called wideband Delphi, in which a group of people choose estimates independently, compare and discuss their individual results, and then revise their estimates to arrive at a consensus.

In planning poker, each card in the deck is marked with a different number of story points, and together the pack represents a rating scale. This scale isn't always even, or linear, though.

It's useful to use small intervals in numbers of story points only when comparing small stories – such as if one story is valued at 2 points and another, involving twice as much effort, at 4 points.

For larger stories, it's better to use even larger intervals. For instance, it's unlikely to be useful to assign one story 20 points and another 21 or 22 points because the values are so similar. Instead it would make sense to assign both stories the same number of points. You might then assign a story that clearly requires much more effort 40 points.

The Fibonacci series is a famous series of numbers in which each number is the sum of the previous two numbers. This results in increasingly larger intervals between numbers as the series progresses, such as the numbers 1, 2, 3, 5, 8, 13, 21 and so forth.

The sequence used to mark the cards in planning poker is typically a variation on the Fibonacci series in which the value 20 used in planning poker would typically be a value of 21 on the Fibonacci scale.

The cards in the planning poker deck are each labeled with a unique value. The most common values in the series are a question mark, 0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40, and 100, as well as a card labeled with the infinity symbol. Each value represents a number of story points.

Although it's popular to use a scale based on the Fibonacci series, you can also choose to use either linear or binary scales.

Graphic

The numbers on the scale are 1, 2, 3, 5, 8, 13, and 21.

In a linear scale, the interval, or increment, between each pair of consecutive numbers is the same – for example, numbers in the series of 0, 1, 2, 3, 4, and so forth. When estimating story points, a team can choose to group sets of ratings, such as values one to four, and seven to nine, to indicate similar or related amounts of effort.

Graphic

The numbers on the scale are 1, 2, 3, 4, 5, 6, and 7.

A binary scale includes the values 1, 2, 4, 8, 16, and 32. Using binary is ideal for categorizing user stories based on varying degrees of effort, because the gaps between the higher values are relatively large.

Planning poker meetings are time-boxed to allow approximately two to three minutes of discussion time per user story. Thereafter, each member of the development team uses numbers from their cards or scale of choice, to choose a reference point for their story point estimation. They then discuss the story, the individual estimations, check for consensus, and then repeat the process to revise their estimates until a consensus is reached.

Select each step in the process to learn more about it.

Choose a reference point

As the first step, the participants – ideally including all members of the project team – should choose a reference point story. Usually, this should be an average-sized story that allows the team to characterize the remaining stories as either smaller or larger.

This user story can then be used as a reference point for estimating the sizes of other user stories. Any user story that requires more work to develop than the reference user story should be assigned a higher story point value, and smaller stories should be assigned proportionately lower values.

Discussing the story

Once they've assigned points to a story for reference purposes, the development team and product owner, who represents the customer's interests, discuss the information about a new user story they want to estimate for no longer than two to three minutes. Developers may ask the product owner questions about the user story, although only at a high level. The aim of this discussion is to help participants decide on their estimates for the story that's being discussed.

At this stage, the participants should refrain from indicating which card – representing a particular story point value – they're thinking of choosing.

Decide on estimates individually

Once participants have finished discussing a user story, each person decides on an estimate for the story and selects a card marked with this number from his or her deck of cards.

Next all the participants in the meeting raise their cards or turn them face up on the table. Everyone should show their cards at once to prevent any participants from changing their estimates based on what other people have decided.

The product owner, who represents the customer's interests, doesn't typically choose an estimate. However, this person's contributions during discussions may include clarifying stories and answering questions for those on the development team.

Check for consensus

Once estimates have been revealed, the product owner and team leader check the cards for consensus in the estimates.

Repeat the process

If there's no consensus, participants return their cards to the deck. The participants with the highest and the lowest story points are asked to give a rationale for their estimates and all participants discuss the user story again.

All participants are then asked to choose new estimates for the user story. They reveal the cards they've chosen, and the process is repeated.

Reach a consensus

The development team repeats the estimation process until all participants reach agreement on the story point value for a user story. This value is recorded, and the team then moves on to discuss the next user story.

The number of participants at a planning poker meeting ideally shouldn't exceed ten people, but if more than ten people are present the participants can be split into teams. Participants are the developers on a development team and can include programmers, database engineers, analysts, testers, and designers, for example.

A facilitator should guide participants through the planning poker process, especially if the team is new to this technique. Anyone who understands the process can be the facilitator – it doesn't have to be the project leader.

Consider an example of how a development team uses planning poker to estimate the story points for a user story or theme. First the facilitator hands each participant several planning cards.

Graphic

The cards have these values: a question mark, 0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40, 100, as well as the infinity symbol.

The facilitator reads out the description of the user story – in this instance, "As a site visitor, I want to be able to browse a collection of books online." The product owner then answers any questions from the development team, and the facilitator asks participants to choose cards they feel represent the story's relative size.

Once two to three minutes have elapsed, the facilitator asks all the participants to reveal the cards they've chosen at the same time.

Each planning poker participant puts the card that they've chosen face down on the table. When you ask them to reveal their choice, all the participants turn their cards over all at once.

The participants have chosen cards with the story point values of 2, 3, 5, and 2, respectively.

As part of facilitating the discussion, you ask the developers who chose the highest and lowest values to explain their choice. This prompts a general discussion about what the user story involves.

After the discussion, you ask participants to vote again in an attempt to reach consensus.

The choices resulting from this round are much closer together – the participants voted for story point values of 2, 3, 3, and 2.

The team then discusses the results and agrees to assign the user story 3 points.

Question

As the facilitator for a planning poker session, you've given each of six team members cards labeled 1, 2, 3, 5, 8, 13, and 21. Each team member chooses a card indicating the number of story points to assign to a particular user story. The resulting values are 3, 5, 8, 8, 13, and 21.

What should you encourage the team to do next?

Options:

1. Try using a linear sequence of numbers instead of numbers in the Fibonacci series
2. Ask the team members who chose the values of 3 and 21 to explain their choices
3. Calculate the average number of points and assign it to the user story
4. Decide what story point value to assign to the user story on behalf of the group

Answer

Option 1: *Incorrect. The particular rating scale that's used won't affect the participants' estimates of how much effort it will take to develop a user story. It also won't affect the level of variation in different team members' estimates.*

Option 2: *Correct. If participants haven't reached consensus about how many story points to assign to a user story after the first round, the facilitator should ask those who chose the lowest and highest numbers of points to explain their reasoning. This may encourage other participants to revise their estimates. The facilitator should also encourage a general discussion and then ask participants to once again choose estimates, repeating the process until they come to a consensus.*

Option 3: *Incorrect. Using the poker planning technique, you encourage participants to continue revising their estimates until consensus is reached. You don't simply average the results of individuals' estimates. Those who have chosen higher or lower estimates may have good reasons for having done this, and can discuss these with other participants so that everyone arrives at the best possible estimate.*

Option 4: *Incorrect. The facilitator guides the process but doesn't make decisions on behalf of the group. Instead the facilitator should encourage participants to discuss their scores and then choose new estimates, repeating the process until they reach a consensus.*

Correct answer(s):

2. Ask the team members who chose the values of 3 and 21 to explain their choices

3. Estimating using ideal days

Instead of using story points, agile teams may estimate the relative sizes of user stories using *ideal days*. This involves estimating how many days it will take a single developer to build, test, and release the functionality described by a user story under ideal circumstances, and relative to the estimates for other user stories in a project.

Ideal days differ from the actual, elapsed time it's likely to take to complete tasks. This is because ideal estimates disregard likely delays, waiting time, and distractions. They also don't account for the possibility of multitasking, or working on more than one task at a time.

In practice, multiple developers may work on developing a user story at the same time. It's also accepted that the actual time it will take to complete a user story will be affected by unforeseen factors that may result in interruptions or delays.

Even a well-planned release typically includes discrepancies between elapsed time and estimated, or ideal, time. For example, you may have estimated that a user story will take five ideal days to complete. But if a developer falls ill, the story may take twice as long.

Consider a team that estimated it would take a developer two days to design an interface for a web page. Matt, the developer who was assigned the task, was unexpectedly asked to present a design workshop for trainees on the day he intended to start work on the interface. This set him back half a day.

On the second day, Matt's daughter fell ill at school and he had to leave to take her to the doctor. By the third day, Matt had barely started working on the web page. Despite the developer's best efforts, the task estimated at two days is now likely to take four actual days or more to complete.

Question

An agile team develops an estimate that's expressed as a number of ideal days.

What does this involve doing?

Options:

1. Estimating the number of days it will take to finish testing a user story
2. Estimating the optimum duration of a project iteration, in number of days

3. Estimating the relative size of a user story by deciding how many days it will take to complete
4. Determining a reasonable deadline for finishing development of a user story

Answer

Option 1: *Incorrect. Using ideal days, a team estimates how many days it will take a developer to build, test, and release the functionality described by a user story, given ideal circumstances. So not only testing time is estimated.*

Option 2: *Incorrect. An agile team uses ideal days to estimate the relative sizes of user stories, rather than to decide on a suitable iteration length.*

Option 3: *Correct. Using ideal days, a team estimates how many days it would take a single developer to complete a user story under ideal conditions, in relation to the length of time it's estimated other stories will take to complete.*

Option 4: *Incorrect. Using ideal days, a team estimates how many days it will take a single developer to complete a user story under ideal circumstances. This type of estimate shouldn't be used as the basis for determining actual deadlines because it doesn't account for likely delays and obstacles, or reflect the fact that multiple developers may work on the story at the same time.*

Correct answer(s):

3. Estimating the relative size of a user story by deciding how many days it will take to complete

Summary

When planning for a project's release, an agile team may use story points to estimate the relative size of each user story, in terms of the amount of development effort it involves.

A popular technique for estimating story point values is known as planning poker. This involves giving each team member a set of cards representing different possible numbers of story points. The participants each choose a card for a user story, reveal and discuss their choices, and then repeat the process until they're able to reach a consensus.

Instead of using story points, a team may use ideal days to estimate user story sizes. This involves estimating how many days it will take a single developer to complete a user story under ideal circumstances.