Transcript

Estimating Velocity

Learning Objectives

After completing this topic, you should be able to

- *estimate velocity for a given team*
- *identify factors that can negatively impact the velocity of an agile team*

1. Estimating a team's velocity

After an agile team has estimated the sizes of user stories, the next step in charting out the release plan is to determine an estimate of the team's *velocity*. Velocity is a measure of team productivity – or, more specifically, it indicates how much work the team can complete in a given iteration. Team velocity is determined based on how many story points or ideal hours it can complete in the time-boxed duration of an iteration. Once a team knows its velocity, it can plan how many units of work – and which user stories – should be assigned to each iteration.

In physics, the mathematical equation for calculating a moving object's velocity is distance divided by time. For an agile project, however, you measure team velocity as actual development effort divided by project time.

Most teams measure their velocity - or rate of progress - in terms of story points completed per iteration, but ideal time can also be used for the calculation.

Consider a development team who has 48 weeks in which to finish developing a new web site in time to meet the customer's release deadline. So the team is trying to establish what their velocity needs to be given the amount of project work for this particular project time frame.

Using historical data from another project, the team is able to estimate their expected velocity on the new project. The historical data revealed that they were able to complete 192 story points in 24 iterations on the last project, so their velocity would have been 8.

Question

An agile team with six developers uses iterations that are each three weeks long. The team estimates it can complete 12 story points per iteration, and the total project duration is scheduled at 36 weeks.

Identify how to calculate the team's velocity.

Options:

1. The number of ideal days divided by six
2. The total work effort measured as story points divided by the number of iterations
3. The number of weeks in the iteration multiplied by six
4. The amount of developer time multiplied by the number of weeks in each iteration

Answer

Option 1: *Incorrect. You calculate velocity by dividing the amount of work effort for the whole project – measured as a number of story points or ideal days – by the number of iterations, not the number of people in the development team.*

Option 2: *Correct. You calculate a team's velocity by dividing the total project work measured as a number of story points – 144 – by the number of iterations – 12. So in this instance the team has an estimated velocity of 12.*

Option 3: *Incorrect. You should calculate velocity by dividing the amount of work effort – measured in story points or ideal days – by the number of iterations used during the project.*

Option 4: *Incorrect. Velocity is measured as work effort divided by project time. So you can use ideal days or story points and divide it by the number of iterations, but the number of team members is irrelevant.*

Correct answer(s):

2. The total work effort measured as story points divided by the number of iterations

You can estimate a team's initial velocity in one of three ways:

- using historical values
- running an iteration, or
- using forecasting

Using historical values is the most practical way to estimate velocity in certain situations. For example, if a development team already has experience with the technology, tools, product owner, and functioning as a unit, you can use the actual velocity it achieved in the past to estimate its velocity for a new project.

However, previous values may no longer apply if the team's dynamic has changed drastically or if the team is expected to operate in very different context from in the past.

You can use historical values to determine a rough estimate of a team's velocity. Then you can revise this estimate after the team has completed a couple of iterations on the current project, using the team's actual velocity during those iterations to adjust the initial estimates and schedule.

An ideal method for determining team velocity is to actually have the team start development for one or more iterations. This has multiple benefits, including a realistic estimate to be used to create the remaining release plan.

It's generally a good idea to adjust the estimate you calculate in this way slightly upward because a team's velocity tends to increase as a project progresses and team members get used to working together.

You can use a test or practice iteration to refine another type of velocity estimate by incorporating your observed data with the other data.

For example, you might use historical data to estimate that a team can complete roughly 12 story points per iteration. However, once you've completed the test run consisting of three iterations, you realize the team has a higher velocity of almost 15 points per iteration.

Graphic

The graphic shows a four-column table that contains three rows. Each row represents an iteration in the test run. The column headers are: Number of user stories, Initial estimate of story points, and Actual complete number of story points. The first iteration has the values 4, 12, and 14. The second iteration has the values 6, 12, and 15. The third iteration has the values 3, 12, and 15.

Once the first iteration has passed, you realize that your team's level of experience and familiarity with working together as a unit has made them more effective than you anticipated. They've completed 15 story points – 3 points more than your estimation.

As the product owner, you can take this velocity information and update the release plan accordingly. You may also decide to adjust the user story allocation to accommodate the additional 3 points of team capacity per iteration.

Sometimes it's not feasible to use historical values or run iterations to estimate velocity. This applies if, for example, a project isn't due to start for another six months, the resources are not yet available, or you aren't authorized to conduct a test iteration until the customer signs a contract. In these cases, you should use forecasting to estimate the team's velocity.

Forecasting can be a time-consuming process, because it involves expanding user stories into tasks, estimating the work effort involved in completing those tasks, calculating the amount of work that can fit into an iteration, and then determining the velocity required to produce the required work in the available time.

Because the forecasting method involves a very high degree of uncertainty and can be time consuming, you should use it only if it's not possible to use historical values or run a test iteration.

Another downside to using this method is that it forces you to break down the work involved in a project – for example, dividing user stories into tasks – in more detail than you normally would at the release planning stage, given an agile approach.

Question

You're responsible for estimating the velocity of an agile team who is about to start work on a new project. The team consists of five people – a product owner, software architect, designer, software tester, and a business analyst. The team is so adept at working together that the software architect in the team sometimes doubles as a software developer.

You're unsure what velocity to assign, given the fact that one team member sometimes performs two roles, so you decide to pick a random number – 15 – and you assign it as the team's velocity.

What would be a better approach to estimate velocity in this situation?

Options:

1. Run an iteration
2. Use historical values
3. Forecast the velocity

Answer

Option 1: *Incorrect. Although running a test iteration is an acceptable method of estimating a team's velocity, it is preferable to use historical values instead.*

Option 2: *Correct. Using historical values is the easiest method to estimate velocity, and can be quite accurate if an agile team has prior experience in working together on a very similar project.*

Option 3: *Incorrect. You should use forecasting only as a last resort, such as if a team has no experience working together. Instead, you should use historical values to estimate the team velocity.*

Correct answer(s):

2. Use historical values

2. Factors that impact velocity

It's no surprise if a development team's velocity varies slightly from one iteration to the next. Velocity remains an estimate and is likely to fluctuate.

Factors that can negatively impact a team's velocity include burnt out or overworked developers, changes in team structure such as if people join or leave a team, and content that's technologically complex. Other factors can include miscommunication, such as when misunderstandings occur between team members, and sudden changes in customer requirements that put pressure on the team.

Select each factor to learn more about how it can affect velocity.

Burnout and overwork

If a development team has been under a lot of consistent pressure, or if team members have been working a lot of overtime, the team may suffer from burnout. Tired people are more likely to make mistakes, which then have to be corrected and cause delays. This can reduce a team's velocity.

Changes in team structure

Changes in a development team's structure – such as new members joining the team or other people leaving – can decrease a team's velocity.

People who have experience working together as a cohesive unit are likely to work faster than people who aren't familiar with each other's skills and habits.

Technological complexity

High or unanticipated levels of technological complexity in the work a team needs to complete is likely to reduce the team's velocity. Similarly, if a team is required to start using a new development tool, language, or database technology, it will have to spend some time adjusting instead of producing.

Miscommunication

Miscommunication or communication breakdowns negatively impact a team's velocity. This can include communication problems among team members or between the team and the customer, or product owner.

For example, if the product owner is unclear when outlining requirements or doesn't provide sufficient detail, it may lead to unnecessary rework and slow down the team's production.

Changing customer requirements

Changing customer requirements can decrease a team's velocity, especially if they mean that a team has to make adjustments to the user stories it's developing mid-iteration.

For instance, if a customer introduces an urgent requirement for a new product function, it might require that the team acquire new skills by introducing a new team member or send another member off to receive training. This may upset the team's balance and impact its velocity.

A project team is currently developing a web site for a shoe retailer. Based on both historical data and the team's performance during the first two development iterations, you estimate the team's velocity as about 20 story points per iteration.

In week two of the current four-week iteration, the customer asks that the team change its focus and work on developing a new feature, based on what a competitor's site is offering. The team, which was mentally focused on producing a different deliverable, then has to adapt.

In addition, programmers have to work a considerable amount of overtime to complete the required feature in time. This makes them prone to errors. It also makes it difficult for them to communicate properly with other team members, who work only during regular hours. The result is that instead of producing 20 story points of work by the end of the iteration, the team barely succeeds in completing 15 points.

Question

An agile team has an estimated velocity of 10 story points per iteration. During the first project iteration, it managed to stay ahead of schedule. However, in the second iteration, a programmer realizes that a particular task is more complex than originally expected. A few days later, a senior web designer leaves the team due to ill health.

Which factors have negatively impacted the team's velocity?

Options:

1. Miscommunication among team members
2. Poor estimation during planning of effort required
3. Changes in the team's structure
4. Changes in the customer's requirements

Answer

Option 1: *Incorrect. There's no indication in this example that poor communication has slowed down the team.*

Option 2: *Correct. A task that proves more challenging than anticipated is likely to require more time to complete. So the extra technical complexity involved may negatively impact the team's velocity.*

Option 3: *Correct. A senior web designer leaving the team will result in a change to the team's structure. Either an existing team member will have to pick up that person's work or a new person may have to join the team. This is likely to have a negative impact on the team's velocity.*

Option 4: *Incorrect. There's no indication in this case that the customer's requirements have changed.*

Correct answer(s):

2. Poor estimation during planning of effort required
3. Changes in the team's structure

Summary

You can estimate a development team's rate of work completion – or velocity – in terms of the number of user stories or ideal days of work it can complete per iteration. For release planning, you can estimate a team's velocity based on historical values, by running an iteration and observing the velocity, or by using a forecasting technique.

Factors that negatively impact a team's velocity include overworked team members, changes in team structure, high or unexpected levels of technical complexity, miscommunication, and sudden changes in a customer's requirements.