How Measuring Velocity Helps in Your Agile Journey

written by Manoj Khanna | September 16, 2014



Measuring velocity is a useful way of determining how long an agile project will take to complete, by providing a rough estimate of the amount of work the team can complete in a sprint. It is necessary to keep in mind, however, that as insightful as knowing your team's velocity is, velocity is *not* a true measure of your product's development.

This post will discuss what velocity is and the reason why we measure it, and why management is interested in higher versus lower velocity.

Velocity Defined

Velocity, in terms of agile product management, is a metric that provides insight into approximately how much work a given team can complete in a sprint. Measuring velocity uses information from a completed sprint, so if your team is approaching its first sprint, you must know how many people will be involved in the project, the maximum amount of work each person can complete, and the total number of workdays in the sprint, in order to calculate the velocity.

Calculating velocity involves units of work that can be defined as hours or story points – a metric capturing the complexity of implementing a story – for example, and tasks. Velocity is calculated by adding up the difficulty metric of

every backlog task, such as stories, completed by the team in a given sprint with the units of work for those tasks. This provides the velocity in terms of units of work per sprint.

Why Measure Velocity?

Velocity provides an estimate of how much work your team, specifically, can complete in a given sprint. As velocity is calculated using the work of previous sprints and if everything remains constant – such as the same people are involved and the tasks are relatively of the same nature – then velocity is consistent. If the team had a velocity of 30 story points per sprint for previous software development projects, then it can be expected that, all else constant, the team will have a velocity of 30 story points per sprint in the next software development project.

Velocity is useful in estimating the approximate length that a project will take, as well. If the project at hand has 120 story points' worth of stories, and previous sprints show that the team has a velocity of 30 story points per sprint, then it can be expected that the team will complete the present project in four sprints.

Higher Velocity Versus Lower Velocity

As velocity signifies an agile team's productivity – the velocity value indicates that either the team completed fewer high-difficulty stories, or they completed many stories of lower difficulty – it is more desirable for a team to have a higher velocity than a lower one. A higher velocity would indicate that the team is capable of completing more stories or high-difficulty stories, which translate into more progress towards the project.

It is worth noting, though, that when teams are measuring their velocity at the start of a project, the velocity value will not be as accurate as it will be for later sprints. An underestimation when initially measuring velocity can result

in a higher velocity later on in the project. Similarly, an overestimation in initially measuring the velocity will lead to a lower velocity later on in the project, as the velocity is adjusted with each completed sprint. As such, the velocity value that is calculated later on in a project may not accurately reflect the team's productivity; their productivity could have remained constant, yet the change in velocity value may be due to an under or overestimation.

Velocity Is Not a True Measure of Product Development

As useful as measuring velocity is, an agile team should not rely on their velocity to indicate the progress of their product's development. The velocity is measured based on each sprint, and with each new sprint, new product features' information is accumulated and added; this changes the stories and story points associated with the next sprint, so the previous sprint's velocity is not necessarily indicative of the team's work in the next sprint. Many new features may have to be added or changed in the next sprint, so the nearly completed product of the previous sprint may actually be 40% completed, given the previous sprint's feedback. Even though the team had a high velocity in the previous sprint, product feedback necessitated even more work before the product can be deemed complete.

{Photo courtesy: Flickr/Jan-Hendrik Palic}