Velocity-Driven Versus Commitment-Driven Sprint Planning

There are different opinions for conducting sprint planning. Long debate is happening between velocity driven sprint planning and commitment driven sprint planning. Mike Cohn, founder of Mountain Goat Software, shared his views in his recent blog on Why I Prefer Commitment-Driven Sprint Planning.

Mike described both the approaches in his blog. Velocity-driven sprint planning is based on the premise that the amount of work a team does in current sprint is roughly equal to what they've done in prior sprints.

The steps in velocity-driven sprint planning are as follows:

- Determine the team's historical average velocity.
- Select a number of product backlog items equal to that velocity.

In commitment-driven sprint planning, a team commits to one product backlog item at a time by roughly identifying and estimating the tasks that will be involved, and stopping when they feel the sprint is full. As per Mike:

> *Because "commitment" is often mistaken as "guarantee," this approach is becoming more commonly referred to as capacity-based sprint planning.*

Mike says that there is no need to deep dive into all the tasks corresponding to one user story at the time of planning.

> *Do not ask or expect a team to think of every task that will be done during the sprint. Not only is that impossible, it is also unnecessary.*
>
> *Teams should think of enough of the tasks that they feel they have thought through the work —but it is important to realize that thinking through the work is the real goal of this meeting. Identifying tasks and hours is secondary.*

Mike says that he prefers commitment driven sprint planning because of the following reasons:

- Velocity is variable so good for long term planning.
- Anchoring, tendency for an estimate to be unduly influenced by earlier information.

> *If you are already doing velocity-driven sprint planning and it's working for you, don't switch. However, if your team is new to Scrum or if your team is experiencing some of the issues, then I recommend using commitment-driven sprint planning.*

Capacity driven planning yields better results for new teams who have not run a sprint together before, because the empirical evidence of task estimation is stronger and usually based on something everyone in the team has done.

Thomas Henson, shared his views on Diqus as:

> *I would agree with your assessment. The key is the tendency for teams to become in keeping the same number of story points per sprint. Commitment-driven allows for the team to challenge themselves. The second point is commitment-driven allows your team to self manage their workloads, and a key of Scrum is for team to be self organizing.*