Rolling Lookahead Planning

by Mike Cohn • 13 Comments • originally published in Iterations on 2008-07-09

Agile processes replace some of the upfront planning of a traditional process with a just-in-time approach in which plans are progressively refined. For the often-used example of a team working in isolation, progressive refinement usually takes the form of creating a high-level release plan and then planning each iteration as it starts. While this is often sufficient for the lone team working on a desert island, it will generally be inadequate on a project being built by multiple teams. When a team needs to integrate its work with the work of other teams, all teams involved will usually benefit from what is known as rolling lookahead planning.

A team doing rolling lookahead planning will begin its iteration planning meetings just like any other team. Usually this is by doing commitment-driven iteration planning, which takes most teams two to five hours to plan a two- to four-week iteration. This is because commitment-driven iteration planning requires the team to determine which product backlog items it will commit to deliver during the iteration by carefully considering each. Product backlog items are discussed and considered in roughly the priority order established by the product owner or customer. The tasks necessary to complete a product backlog item are identified and estimated. When the team finishes discussing the work involved to complete a product backlog item, team members decide whether they can commit to completing it in the coming iteration.

Once the current iteration has been planned, the team remains in the iteration planning meeting and plans two more iterations. Wait, before you panic let me point out that while it often takes two to five hours to plan the first iteration, this is because the second and third iterations are planned at the user story or product backlog item level rather than with tasks as was done with commitment-driven iteration planning for the current iteration. Using data such as its historical average velocity, or a projection of future velocity if there’s a good reason to deviate from history, the product owner and team identify the product backlog items that are likely to be developed in the next two or so iterations.

The team is not committing to deliver these more distant items in the same way it is committing to complete the product backlog items of the current iteration. Specific commitments will be
made to those iterations when each rolls forward to become the current iteration. Until then, each is planned only at a high level and only by identifying the work likely to be done in each iteration.

A team does not perform rolling lookahead planning to identify design considerations for the current iteration, although that may occur and be a side-benefit. The chief purpose of rolling lookahead planning is the coordination of work to be done by multiple teams. As new iterations roll into the planning horizon, teams look ahead to the functionality likely to be developed in those iterations. In doing so, teams on a large project are likely to find dependencies upon one another.

For example, suppose both your team and mine are ready to plan the fourth iteration of what will be a seven-month project. We conduct separate meetings, committing to features to be developed in iteration four. We separately identify and estimate the tasks to be performed in that iteration. Before concluding our separate meetings, our teams each use historical velocities to forecast what they will do in iterations five and six.

After both iteration planning meetings are over, you tell me that your team has identified a dependency on my team. You need me to develop something and pass it along to you so that you can complete a particular product backlog item. I tell you that my team cannot do it during iteration four as we just finished planning that iteration. That’s fine with you because you don’t need it until iteration six as it just now rolled into view. We agree that my team will develop the requested functionality in iteration five, delivering it to you at the start of iteration six. By looking ahead two iterations we have successfully avoided the emergency of “my team needs this from you this iteration” that can otherwise be common on large agile projects.

Rolling lookahead planning, and specifically looking ahead two iterations, is a useful technique that belongs in the toolbox of any agile team working on a large project.

Posted: July 8, 2008
About the Author

Mike Cohn specializes in helping companies adopt and improve their use of agile processes and techniques to build extremely high-performance teams. He is the author of User Stories Applied for Agile Software Development, Agile Estimating and Planning, and Succeeding with Agile as well as the Better User Stories video course. Mike is a founding member of the Agile Alliance and Scrum Alliance and can be reached at hello@mountaingoatsoftware.com. If you want to succeed with agile, you can also have Mike email you a short tip each week.