

8 steps to kick off quarterly planning

Learn how to build great software using long-term agile planning



BY CLAIRE DRUMOND

BROWSE TOPICS

- [Agile manifesto](#)
- › [Scrum](#)
- › [Lean portfolio management](#)
- [OKRs](#)
- **Long-term agile planning**
- [What is SAFe?](#)
- [Spotify model](#)
- [Organizational agility with Scrum@Scale](#)
- [Scaling agile with Rosetta Stone](#)
- [Agile iron triangle](#)
- [Using Improvement Kata to support lean](#)
- [The Large-Scale Scrum \(LeSS\) framework](#)
- [Beyond the basics whitepaper](#)
- › [Software development](#)
- › [Design](#)
- › [The agile advantage](#)
- [DevOps](#)
- › [Agile Teams](#)
- › [Agile tutorials](#)
- › [About the Agile Coach](#)
- [All articles](#)



Going into another quarterly planning meeting at work, I realized that I too am working on a long-term project. I'm building a house. Building software and building a house aren't all that different – both are long-term projects where multiple teams need to coordinate with one another, and any homeowner will confirm, the project is never done. Improvements can always be made, something always breaks, and market trends always change. Without a plan, you're at risk for potential for blockers or a move-in date that's always two months away.

Where software development differs, and home building could definitely benefit, is the use of agile methodology. Agile allows multiple teams to respond to changes, quickly. So how can agile, a method based on frequent, continuous delivery exist with long-term, big-picture planning? Is it possible to create a realistic forecast over a long period of time, knowing that the one constant is change?

How long-term planning and agile work together

No matter where you are on your agile journey, whether you're applying kanban, or scrum, or are just starting to dabble with [agile at scale](#), there's still a need to manage [people, work, and time](#) when [planning your long-term strategic vision](#). In software

RELATED READING

[The iron triangle of planning](#)

[Learn more →](#)

SUBSCRIBE

[Sign up for more articles](#)

Email

[Subscribe](#)



WHITEPAPER

Beyond the basics of scaling agile

Learn the new rules to transformational agility.

[Download the whitepaper →](#)

planning your long-term strategic vision. In software, it's hard to paint a vision with disconnected tools, like Gantt charts, spreadsheets, and custom mixes of project portfolio management (PPM) tools. Or in my contractors case, his spreadsheet, email, and text message mashup becomes instantly impractical.



The key to long-term agile planning is keeping your project details and task estimates in sync with your roadmap.

Before we talk about dynamic forecasting solutions, let's talk about the steps to build a long-term agile plan using the metaphor of building a house:

Step 1. Start with the big picture.

Whether it's a house or a product, you need to define the vision and outline the strategic themes. Think of themes as organization-wide focus areas. What do you want to focus on over the next quarter, 6-months, year? Where do you want to spend time and resources? Performance, user experience, security, new competitive features (hot tub anyone?), or a combination of all these?

Of course, I wanted everything, but there's always those two pesky realities – time and money. Setting your high-priority themes will help you focus your time and energy to do a few things really well.

Step 2. Identify the big ticket items.

For example, to support the security theme, we'd need to build a new foundation, with sheer walls on the frame, solid-core doors, and double-pane windows.

Step 3. Break it down.

So what work actually goes into installing new, secure, windows? Break down the work for the larger initiative into more consumable pieces, like [epics](#). This will give you a granular view into all the steps required to achieve your initiative.

For example, you have to remove the old windows, buy new windows, install them, improve window experience with coverings. These pieces of work would populate your [backlog](#).

This will help you with the next, most important step in this planning process: estimating.

Step 4. Get estimating.

After work is broken down into chunks, you'll need a rough estimation of time to make a roadmap. A roadmap is a plan of action for how a product or solution evolves over time. You need this to understand when big things will happen and in what order.

This is where the expertise of your contractor (or product and dev managers) really come in to play. By looking at similar efforts in the past, you can get a good idea of what it takes to complete each epic.

Because estimating requires knowledge of past efforts for similar epics, it makes it even more important to store your information in one place. That makes looking back much simpler and your estimates more accurate. My contractor relies on his experience, but what happens if he forgets, or if a different contractor has a different experience?

At a later date, you'll take the final roadmap to the team (developers, window installers, etc.) where they'll scope it out with even more accuracy.



Step 5. Create smart releases.

When using agile development, teams generally deliver a working piece of software at the end of each sprint as a release (or version). However, when you're long-term planning and roadmapping, you need to define some rough release points on your roadmap, so you can estimate release dates over the next quarter. Like "Outside of the house complete," which combines the secure window initiative, along with framing, paint, insulation etc.

Package the work items in your backlog with features that are similar, make the most sense together, or provide value as a whole to customers. And remember, releases are driven entirely by the scope rather than strict cut-off dates.



Release 1
Exterior complete



Release 2
Interior complete

Step 6. Generate the roadmap.

Now you have an estimated backlog, releases, and teams with a velocity. The traditional planning triangle shows that a plan has three variables: scope (what you want to do), time (how long it will take to do it), and resources (who can do it). You have everything you need to create a realistic roadmap. Finally, your contractor can give you an idea of your actual move-in date!

Pro-tip: This is where you can use a tool like [Portfolio for Jira](#) to create a realistic roadmap for your teams, make data-driven decisions, and keep stakeholders up to date on how your teams are tracking.



Step 7. Share with the team and validate.

Take your new, shiny roadmap, back to your team and validate it. Let the team break down the epics into [stories](#) and give their best estimations for how long the work will take. Your roofers may have some scheduling conflicts, or the foundation company ran out of concrete, so it will take six weeks to order. Use these outside factors to validating your assumptions and make your estimates, and the steps needed to complete your epics, more accurate. You should run your roadmap by key stakeholders, as well, especially if their approval is needed to move forward on certain steps.

Step 8. Keep improving.

With agile, and homeownership, no project is ever done. Continuing to deliver value through incremental

improvements is what drives innovation and a killer home. Use your roadmap to inform and optimize future roadmaps. Get customer, or your family's feedback, and keep testing and improving on a frequent and regular basis.

Interested in building a roadmap to showcase your long-term vision? Atlassian's Portfolio for Jira and Jira Align both offer roadmapping capabilities to support you. [Discover the differences](#) and determine which is the right fit for your business.

SHARE THIS ARTICLE



CLAIRE DRUMOND

Claire Drumond is a marketing strategist, speaker, and writer for Atlassian. She is the author of numerous articles published on the Trello and Atlassian blogs and is a regular contributor to various publications on Medium including Hacker Noon, Art+Marketing, and Poets Unlimited. She speaks at tech conferences around the world about agile, breaking down silos, and building empathy.



TUTORIAL

Learn scrum with Jira Software

A step-by-step guide on how to drive a scrum project, prioritize and organize your backlog into sprints, run the scrum ceremonies and more, all in Jira.

[Try this tutorial →](#)



ARTICLE

The iron triangle of planning

Take a cue from the decades-old iron triangle of planning and learn how balancing different variables can help agile software teams scale.

[Read this article →](#)

Agile Topics

Agile project management

Scrum

Kanban

Design

Software development

Product management

Teams

Agile at scale

DevOps

[Sign up for more agile articles and tutorials.](#)

Email

[Subscribe](#)



Up Next

