

The Benefits of Feature Teams

by Mike Cohn •

27 Comments

Moving away from component teams is a difficult but necessary step for those who want to adopt an [agile project management](#) approach. For example, when I first began to consult for a certain California-based game studio, its teams were organized around the specific elements and objects that would exist in the video game it was developing. There was a separate team for each character. There were weapons teams, a vehicle team, and so on. This led to problems, such as weapons too weak to kill the monsters, colors too dark to show secret passages, and obstacles that frustrated even the most patient player. The studio clearly needed to change its team structure. On more traditional, corporate projects, we see equivalent problems when teams organize around the layers of an application, including

- Reduced communication across the layers
- A feeling that design by contract is sufficient
- Ending sprints without a potentially shippable product increment

If structuring teams around the layers of an architecture is the wrong approach, what's better? Rather than organizing around components, each team on a project can ideally be responsible for end-to-end delivery of working (tested) features. There are many advantages to organizing multiteam projects into feature teams:

- **Feature teams are better able to evaluate the impact of design decisions.** At the end of a sprint, a feature team will have built end-to-end functionality, traversing all levels of the technology stack of the application. This maximizes members' learning about the product design decisions they made (Do users like the functionality as developed?) and about technical design decisions (How well did this implementation approach work for us?)
- **Feature teams reduce waste created by hand-offs.** Handing work from one group or individual to another is wasteful. In the case of a component team, there is the risk that too much or too little functionality will have been developed, that the wrong functionality has been developed, that some of the functionality is no longer needed, and so on.
- **It ensures that the right people are talking.** Because a feature team includes all skills needed to go from idea to running, tested feature, it ensures that the individuals with those skills communicate at least daily.

- **Component teams create risk to the schedule.** The work of a component team is valuable only after it has been integrated into the product by a feature team. The effort to integrate the component team's work must be estimated by the feature team, whether it will occur in the same sprint during which it is developed (as is best) or in a later sprint. Estimating this type of effort is difficult because it requires the feature team to estimate the integration work without knowing the quality of the component.
- **It keeps the focus on delivering features.** It can be tempting for a team to fall back into its pre-Scrum habits. Organizing teams around the delivery of features, rather than around architectural elements or technologies, serves as a constant reminder of Scrum's focus on delivering features in each sprint.

Of course, there will be occasions when creating a component team is still appropriate, for example when a new capability will be used by multiple teams or when the risk of multiple solutions being developed for the same problem is high. Overall, however, the vast majority of teams on a large project should be feature teams. Additional advice on feature and component teams can be found in Chapter 10, "Team Structure," of [Succeeding with Agile](#).

Posted: December 7, 2009

Tagged: sprinting, estimating, teamwork, agile project management, teams

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.
