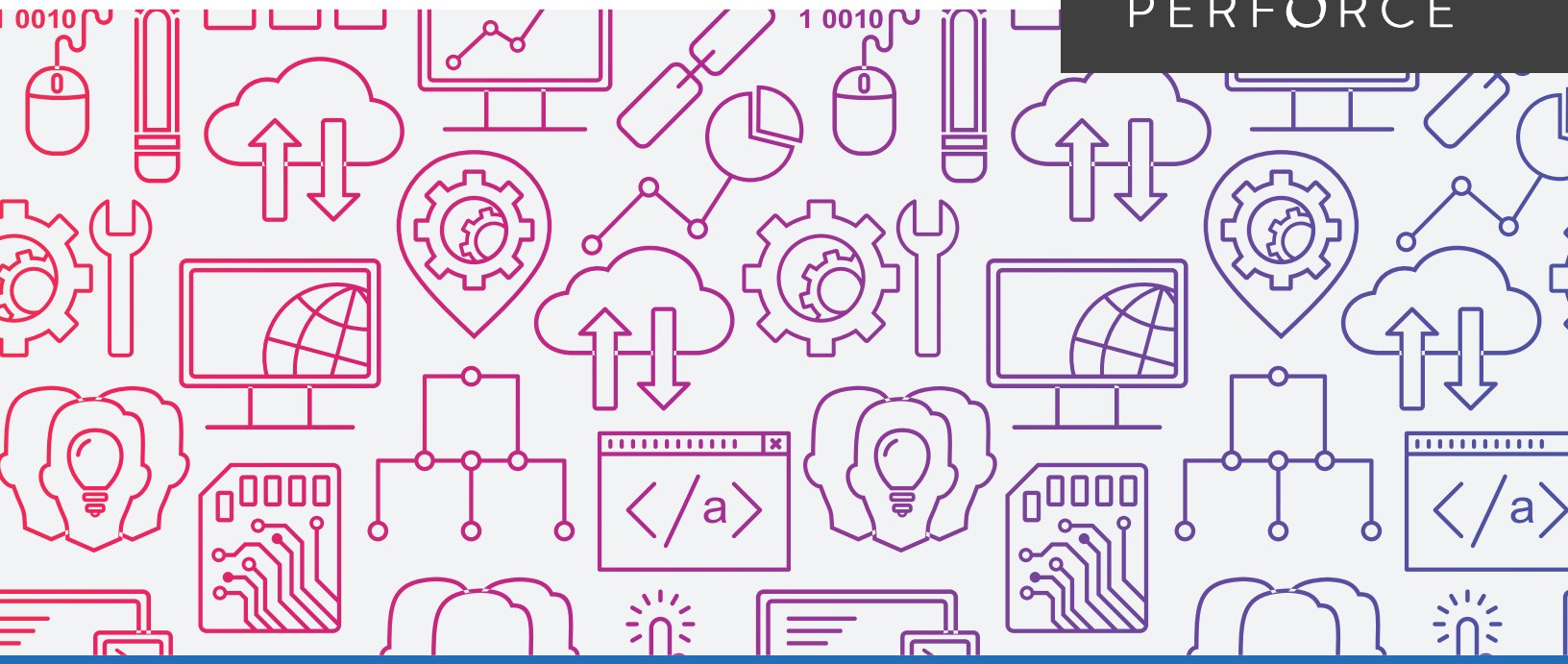
PERFORCE



WHITE PAPER

5 Practices for Reducing Requirements Churn

Requirements churn — or changes to a product's requirements — is inevitable, but there are ways to keep it from becoming excessive. Some churn is natural in the beginning of a project, as the developers refine their understanding of the product and technologies being used to develop it. In most cases, requirements stabilize after they become part of a baseline and the relevant stakeholders have signed off.

If the amount of churn is still high after this point, it can drive up costs, impact quality, or result in sacrificing key features or functionality in order to release the product on time.

Here are five ways to keep requirements churn in check and your project on time and on budget.

1 Use Shorter Release Cycles

One way to limit churn is to work in shorter, more agile release cycles to reduce the scope of each release. The smaller scope in turn reduces the release planning effort because there are fewer requirements to be addressed. As a result, the product gets to the customer faster, decreasing the chance of "out of scope requests." From the customer's perspective, knowing the next release will be coming quickly makes it less risky to defer some features to later releases.

Sometimes churn happens because the team makes educated predictions about the requirements before all of the variables are known. The longer your release planning phase is, the further into the future you're trying to estimate and the more assumptions you have to make — about what has to be done, how it will be done, and even who will be available to do it. Predictions are rarely 100 percent accurate, and the further out you predict, the less accurate you'll be. As a result, the requirements may need much more work or resources than you've estimated.

Again, shorter release cycles can help. Because a shorter release cycle means smaller requirement scope, it's easier to make better predictions. The reduced length of the planning phase means you aren't estimating so far into the future. Because requirements are written closer to the time they'll be turned into code, your predictions are more accurate.

2 Set a Limit, Then Reassess

Another way to reduce churn is to set a limit to the number of versions a requirement can go through after it's considered stable. This will give you an early warning of problems before the project bogs down in churn.

Once the limit is met, gather the key stakeholders and take another look at the requirement to determine the underlying reason for these changes. Ask:

- Was new information discovered about the implementation difficulty?
- Does the stakeholder or customer lack a clear understanding of what the requirements represent, and therefore keeps changing his mind?
- Did new data come to light about competitors, regulations, business deals, or customer feedback?

Once you know what's causing the churn, you'll be in a better position to decide if the requirement is still worth implementing, or if you should delay it for a later release.

Because a shorter release cycle means smaller requirement scope, it's easier to make better predictions. The reduced length of the planning phase means you aren't estimating so far into the future. Because requirements are written closer to the time they'll be turned into code, your predictions are more accurate.

3 Reduce Review Fatigue

The more time it takes for your stakeholders and team to review a requirement, the more likely they are to put it off. When they do finally review it, the chance they'll pay close enough attention to notice specific changes also drops. We call this "review fatigue."

Some teams store all requirements in one large document, and this can be a major contributor to review fatigue. Who wants to wade through 50 pages just to find the 10 requirements they need to review?

You could send team members only the requirements they need to review, but that may not provide the level of detail needed to quickly see the relevant changes. Sometimes you need to refer to the other requirements that haven't changed to properly evaluate the changes you're reviewing.

The faster and easier you can make it for reviewers to see what's changed, the less review fatigue you'll encounter. A requirements management tool can make it easier for reviewers to find the changes they need to address.

The faster and easier you can make it for reviewers to see what's changed, the less review fatigue you'll encounter. A requirements management tool can make it easier for reviewers to find the changes they need to address, with links to provide context if they need it. Such solutions also allow project managers to keep track of who has reviewed a requirement and provided feedback.

4 Improve Visibility

Overlooked requirements may not be discovered until late in the development process, becoming a source of churn. Requirements can be overlooked for a number of reasons — from having many small, concurrent projects in development to a lack of communication between product owner, project manager, development, testing, QA, stakeholders, and management.

In order to prevent missed requirements, you need to ensure all requirements are visible to the team. One way to improve requirements visibility is to use a requirements management tool to centralize access to requirements, allowing for simultaneous collaboration by all stakeholders. Each requirement can be assigned to an individual or team, and then tracked for progress.

5 Expose the Impact of Change

Some churn occurs as the result of team members thinking a change is small and won't have much of an affect on the overall project. Often, "small" changes have a ripple effect that can end up adding weeks to the development schedule.

To prevent this kind of churn, everyone on the team must be able to easily assess and analyze the impact of a change before they request it. Making it

easy to see the impact of a change often helps team members be more judicious about the changes they submit. It also provides you with ammunition for pushback; you'll be able to clearly explain how the proposed change will affect the project schedule.

This is another area in which a requirements management tool can help. Manual impact analysis can be a laborious process, and team members who are pressed for time might guess at how a change will affect the project instead. A requirements management tool makes it simple to generate an impact analysis report for each change, so team members are more likely to actually analyze the impact of a change.

Such tools also allow project managers to track all requirement details, including the change history. By providing a single source of truth, you can avoid the unnecessary churn caused by "small" change requests.

Churn Happens

Everyone wants their project to go as smoothly as possible, but the reality is that some requirements churn is inevitable. By using these five practices, however, you should be able to reduce unnecessary churn and keep your project on time and on budget.

Reduce Churn with Helix RM

Helix RM is a full-featured, modern RM solution from Perforce. With Helix RM, you can easily collaborate with multiple stakeholders, capture requirements, perform reviews, know what's approved, and see the impact of changes.

And Helix RM is even more powerful when integrated with the rest of the Helix ALM suite, automatically maintaining forward and backward traceability without costly manual steps, verifying the test coverage of requirements, and even tracing requirements to source code changes.

Check out Helix RM's full feature set to see if it's the right RM tool for you.
https://www.perforce.com/products/helix-requirements-management

About Perforce

Enterprises across the globe rely on Perforce to build and deliver digital products faster and with higher quality. Perforce offers complete developer collaboration and agile project management tools to accelerate delivery cycles – from agile planning tools to requirements, issues and test management, which then link to all source code, binary assets and artifacts for full build and release tracking and visibility. The company's version control solutions are well known for securely managing change across all digital content – source code, art files, video files, images, libraries - while supporting the developer and build tools your teams need to be productive, such as Git, Visual Studio, Jenkins, Adobe, Maya and many others. Perforce is trusted by the world's most innovative brands, including NVIDIA, Pixar, Scania, Ubisoft, and VMware. The company has offices in Minneapolis, MN, Alameda, CA, Mason, OH, the United Kingdom, Finland, Sweden, Germany, and Australia, and sales partners around the globe. For more information, please visit www.perforce.com