

Why you should be implementing continuous delivery

by Fernando Gonzalez - Tuesday, December 06, 2016

<http://blog.belatrixsf.com/why-you-should-be-implementing-continuous-delivery/>

“Continuous delivery is the way — even if you never achieve it, the mere pursuit will make your organization better.”

Steven Winter, Chief Strategist at GuerrillaQA

Continuous delivery, CD, is the approach driving today’s most successful tech teams in companies like Facebook, Etsy, Amazon and eBay. CD is the coalescence of development, operations and quality assurance (QA) teams in an agile process to deliver software - updates, improvements, fixes, and features rapidly to production.

By making smaller, more frequent incremental improvements, companies can stay ahead of the curve in satisfying customer experiences and innovation, while also improving efficiency, building more cohesive teams and improving company culture.

A few weeks ago we [held a webinar](#) together with Steven Winter of [GuerrillaQA](#) where we examined continuous delivery in detail. In particular we discussed the advantages of CD, which include:

- **Business agility.** CD brings you closer to your customer because it gives you the ability to let them know you are listening, and that you are responsive to feedback. This can show up in simple ways like making a requested color or style available, or refining and clarifying the way you describe a product if it consistently garnered a lot of similar questions.
- **Speed.** CD increases your ability to respond to customer demands and bring new features to market rapidly.
- **Higher quality.** Key to CD practices is building quality in upfront. It means taking the time to do it right. A core part of CD involves embracing a test-first culture, and a higher quality culture, which will help you save costs and move faster. For example, when a developer checks in their code, it should trigger some of the build and unit test framework, so the developer gets immediate feedback.

Continuous delivery unlocks opportunity through:

- **Planning** — Small, frequent releases that build quality in upfront and take advantage of emerging technologies.
- **Integration** — Of your DEvelopment + OPerationS teams, collaboration becomes the standard operation procedure.
- **Testing** — Continuous testing is the core of continuous development. The QA team will need to be much more focused on automation than in the past.

- **Deployment** — Automated software releases in reliable and repeatable processes.
- **Feedback** — Monitoring shifts in your KPIs create processes to integrate what you learned in the next planning phase.

Implementing continuous delivery however does involve tackling some key challenges. One of the most common challenges is cultural. In some cases for instance, developers may take the perspective that there is a QA team for testing, so why should they be writing tests, and spend less time developing? However, in our experience, once developers start with the process, and can see the results of their work immediately, it helps change this perspective.

The most critical element to understand about continuous delivery, is as Steven Winter said in the webinar, just trying to get there. Because even if you never fully achieve it, the journey will make your organization better.

As you start on your continuous delivery journey, we recommend setting a long range goal and begin working in increments that are sustainable for your company. Empower your people and promote teamwork as you integrate your teams. Promote continuous learning that builds trust and transparency on the team and recognizes each individual's contribution.

To find out more about continuous delivery, [listen to our recent webinar](#).