Andrew Zurn

Reading 7

Summarized Article: <http://www.drdobbs.com/architecture-and-design/continuous-delivery-the-first-steps/240161356>

Other Article:

<http://cacm.acm.org/magazines/2013/2/160168-power-challenges-may-end-the-multicore-era/fulltext>

Agile and Lean software development processes have become the new and hip thing in the software engineering world, have recently grown greatly in popularity over the traditional Waterfall method. However, a new software development paradigm has made its way into the headlines, that paradigm being the new “Continuous Delivery” (CE) practice. In his article, called “Continuous Delivery: The First Steps,” Dave Jabs, CTO of AccuRev explains the philosophy behind this new development pattern, and explains how companies might start to integrate it in order to achieve shorter iterations and a massive increase in code deployments.

The philosophy behind CE is best described in Jabs’ words. He states, “the essence of my philosophy to software delivery is to build software so that it is always in a state where it could be put into production. We call this *Continuous Delivery* because we are continuously running a deployment pipeline that tests if this software is in a state to be delivered.” CE thus “leverages automation for testing, software deployment, and promotion… [so] new software can continuously flow, and as it flows, it is validated to progressively higher levels of quality.” In order for organizations to take advantage of CE, Jabs highlights some mains points of reorganization, those being each iteration of a story should become shorter and cycle times should be also shortened, that unit testing should be focused on, with smaller integration tests, and that there should also be smaller teams of developers, each in charge of their own area of an application. In short, if an organization achieves this pattern of development, they could see massive gains in the successful deployment of new builds, and always have new software, and safe code to deploy.

I was definitely interested in this pattern after having heard about it some of the podcasts that I listen to. I also have seen various aspects of it in the agile development pattern that I was able to take part in at Thomson Reuters last summer, as small teams worked on very short iterations in their stories, which caused the business unit I was in to have almost bi-weekly releases to production. In the end, after both reading about it, and having unintentionally experienced this development pattern, I would say it’s something I would hope to be a part of in the future, and think is a potential replacement for even the new patterns of Agile and Lean practices.