Showstoppers for Continuous Delivery in Small Scale Projects

Jakob Svemar
Faculty of Engineering, Lund University

***Abstract*—Small scale software development projects provide their own difficulties.**

This thesis is looking at the possibility of introducing continuous delivery in such an environment.

The solutions are focused on creating an iterative process and including the customer in the early stages.

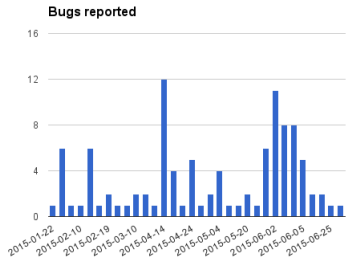
The way software is developed can vary greatly, and different settings create different conditions. A consultant firm can provide a range of services. One of which is in-house development. Their largest team consisting of nine developers, a scrum master and a project manager, is working on ticket solutions for public transportation. This spans several different customers, generally the public sector, and it has been the main focus during this thesis. This has created a situation where projects run in very short time frames with limited budgets. This means projects are set up in a traditional manner and shares a lot of similarities with the waterfall model.

From initial discussions in the beginning of this thesis, two perceived problems were defined. Together with my desire to explore the benefits and possibilities of implementing continuous delivery formed the basis for this master thesis. The problem statements were:

- How do we know when we are done testing?
- We find bugs too late in the development cycle.

Finding the point where the software is stable enough becomes difficult when relying on manual exploratory testing. It takes a lot of time and is prone to human error. Another side effect is that regression testing is reduced to a bare minimum. A previously implemented feature may remain untested through several iterations of software.

If the majority of testing is done at the end of a project, the majority of bugs will be found at the end of the project as seen in the figure.



Bugs reported for the analysed project.

The purpose of this thesis has been to look at the current development practice, identify bottlenecks and figure out changes that could help remove them.

The scope is limited to the in-house development and specifically the team creating ticket solutions. Target audience is developers in similar situation where small projects are common and automated testing is scarce.

Continuous delivery really shines when used in a project with a live product. When developing a product from scratch it becomes a bit more complicated. Still, releases to the customer can be used to evaluate single features to get early feedback before it becomes a base for future features.

To improve the continuous delivery process, or even continuous integration, quality assurance focus needs to change. Relying on manual testing keeps regression testing to the minimal, it also make testing slow and unreliable.

By creating acceptance tests during development, stories will have a clearer definition of when they are done while providing a link between test cases and stories and allow the customer to validate features early on.

Setting up a continuous delivery pipeline is not beneficial enough when applied to only one small project. They need to be reused for future projects and shared with other in-house teams.

The needs for automation should be focused around testing. As the process matures and releases are made more frequent an automated release process will be a logical step.

Author: Jakob Svemar
Contact information: jakob.svemar@gmail.com