Continuous Delivery

Stephen Levitt

School of Electrical and Information Engineering
University of the Witwatersrand

A Definition

> “*Continuous Delivery is a software development discipline where you build software in such a way that the software can be released to production at any time.*”
>
> — Martin Fowler

Development Scenario

Imagine a small software team working on a mobile grocery list application. The application already has the capability of capturing simple shopping lists and checking items off the lists.

- Jeremiah is working from home (remotely) on his own laptop on allowing lists to be shared between users. This feature will take about a few days to implement and he is about halfway through completing it.
- Emma works at the office, using a company PC, and is adding functionality for assigning list items to store aisles. This will take two days and she is about to complete it.
- The application is available in Google Play store.

Questions

1. Why should we even try to achieve Continuous Delivery?
2. What about software quality - how will features be tested if we have to ship at any point?
3. How will we deal with the fact that at any given point in time some features are not complete?
4. How do we get a new feature coded on a developer's laptop into the hands of a user?
5. How often should the team collaborate and integrate their work?

Continuous Delivery/Deployment for Different Types of Software

Consider the following types of software:

- Web applications
- Desktop applications
- Mobile applications
- Embedded software

Discuss the unique challenges that continuously delivering/deploying to these different platforms presents. Which is the easiest to achieve?