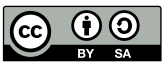
Integration / versioning / deployment



2019-05-06 22:55
1a0f14a

Contents

Waterfall integration

- Each developer would work on the code independently
  - Alice has a copy of the code, she's trying to add a new authentication mechanism.
  - Bob has a separate copy of the code, he's trying to fix bugs reported in a previous version
  - Charlie has a third copy of the code, he's trying to add a reporting mechanism.
- After our 3 devs finish their independent changes, they need to merge them together into the main product. This merge is called "integration".
- Difficulty of integration depends on:
  - How extensive the changes are
  - How much they changes overlap or conflict
  - Both of those are related to how long the changes took.
  - Easier to integrate if we've been working independently for a short time (hours or days) than for a long time (weeks or months).
- If last integration was a long time ago, we end up in "integration hell".

Continuous integration

- Avoid "integration hell" by merging **at least** daily, sometimes merge multiple times per day.
- The "CI" terminology comes from Grady Booch, 1991 book on OOA&D.
- Was promoted heavily with XP = "Extreme Programming" by Kent Beck, early 2000s.
- Relies heavily on being able to automate building and testing.
- We have a "baseline" branch, or a "trunk", or "master" that contains every commit made every day.
- Want the automatic build and test steps to be **fast**.
  - Reduce by parallelizing on a build server, or **build farm**.

  - For libraries, want to build and test for different versions of language, tools, dependencies.
- Existing services/tools: Travis CI, Circle CI, GitLab CI, Atlassian Bamboo.
- Continuous delivery: upload built code to server for users to download.
- Continuous deployment: built code "goes live" on web site immediately from CI service.
- Many CI systems support "badges" to indicate publicly and widely the state of the build, coverage, etc.

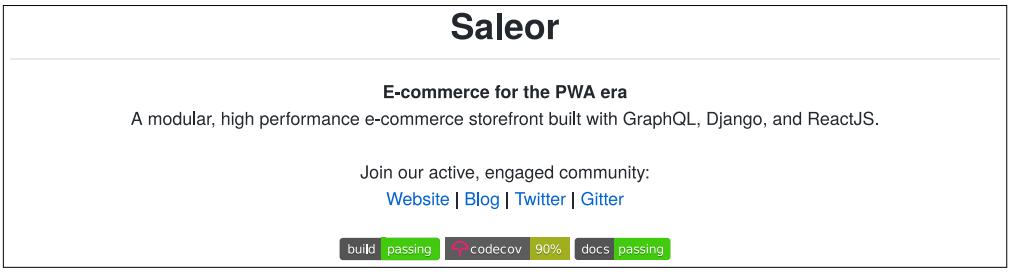


Figure 1: Example of CI badges on a Python project on GitHub

Release management

After integration and testing, next steps include things like tag, bundle, deliver, and/or deploy.

Tagging means assigning the source an "external" version number.

Semantic versioning[1] is a precise way to construct version numbers.

[1] semver.org/

Important to be able to retrieve a version number from within an application. (Also can put git commit ID, timestamp.)

Delivery/deployment has changed over time, and depends on the nature of the project.

1. Shrink-wrap software – CD/DVD in a box with a plastic wrapper.
2. Click-wrap software & app stores
3. SaaS = Software as a Service (web-based software)
4. Custom/proprietary deployments

DevOps…

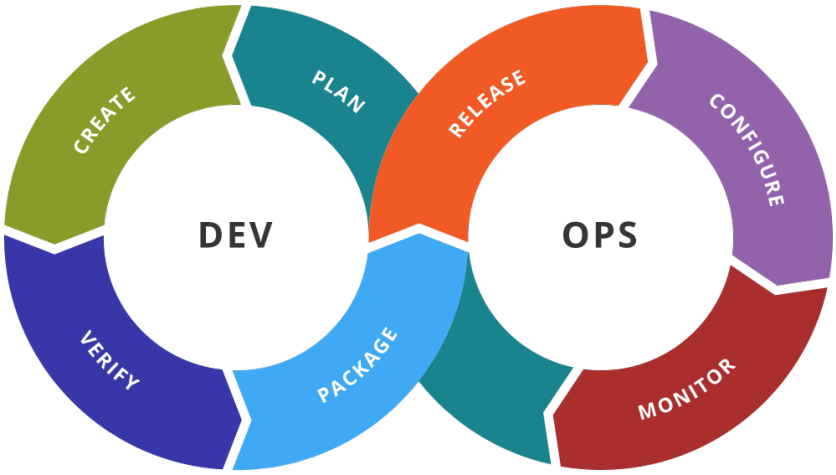


Figure 2: Schematic of DevOps activities

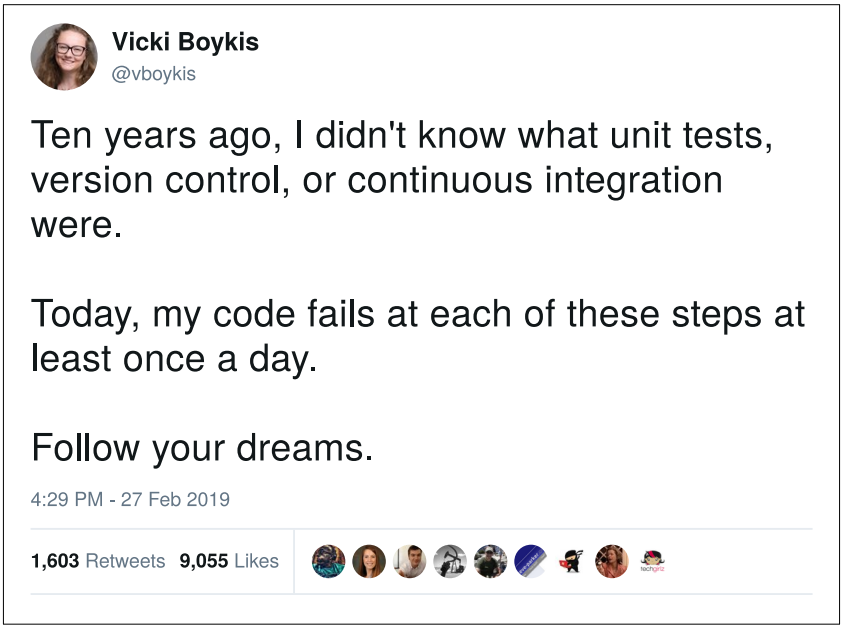


Figure 3: @vboykis on Twitter