Continuous Integration and deployment of applications

by Anton Minashkin

Continuous integration

Continuous integration (**CI**) is the practice, in software engineering, of merging all developer working copies with a shared mainline several times a day. It was first named and proposed as part of extreme programming (XP). Its main aim is to prevent integration problems, referred to as "integration hell" in early descriptions of XP. CI can be seen as an intensification of practices of periodic integration advocated by earlier published methods of incremental and iterative software development, such as the Booch method. CI isn't universally accepted as an improvement over frequent integration, so it is important to distinguish between the two as there is disagreement about the virtues of each

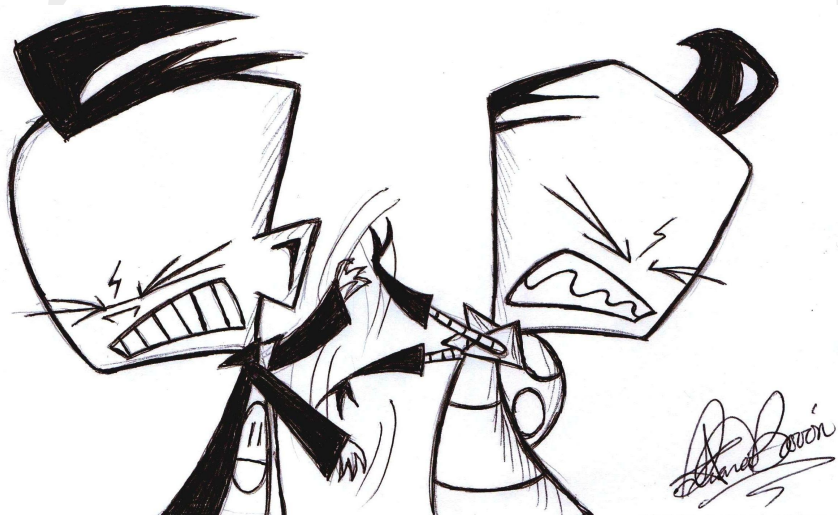
Wikipedia

1. Inconsistent sources

- Someone forgot to commit something
- Build between two commits
- Someone forgot to switch target to “release”

2. Never knows whose build is "the latest one"

Developer A and developer B both have last source version and both have uncommitted changes. Both of them could prepare build.



3. Unit tests? Oh no! I've forget to run tests... :(

It is not enough just to have unit tests (BTW did you have it?)

4. Where our previous build?

You know, this release is really good! But I need the one which we've made couple month ago... Or couple years... Or both of them

5. Who has broke the build?

Why it is always broken? Who did this again? Why today?

Let CI handle them all!

- Bamboo
- Hudson и Jenkins
- CruiseControl
- TeamCity
- BuildBot
- Travis CI

1. Trigger

- Commit
- File system’s change
- Time
- Another build
- “Red button” (manual build)
- Web service was changed

2. Update

Update local sources with new version from VCS

3. Analyse

Static code analyzer

- Typical errors
- Quality of code (size, complexity, etc.)
- Code style

4. Build

Compiling, processing, translating… on build server. No more wasted time on developer's computers.

5. UnitTests

Unit Tests run after EVERY build. Also you could check code coverage on this step.

6. Deploy

Deploy build on a test environment

- Clear and update DB (if needed)
- Deploy on a test server (for web projects)
- Reinstall program (for desktop projects)
- Install on mobile devices (for mobile projects)

7. Test

- Integration tests
- Automated UI tests
- Regression tests
- Functional tests

8. Archive

- Commit tag to VCS
- Save binary files to archive
- Send binary files to testers (or customer)

19. Report

- Change log
- Test logs
- Build log
- Statistic and metrics
- Build reason

CI profits

- Integration without pain
- No more “It works on my machine!”
- Analyze every build
- Run tests on every build

Thanks!

- http://martinfowler.com/articles/continuousIntegration.html
- http://www.jetbrains.com/teamcity/
- http://jenkins-ci.org/
- http://hudson-ci.org/
- anton.minashkin@outlook.com