

Continuous Integration

Martin Kropp[1]
Modified by Na Meng

1

Overview

- Why integration?
- What is Continuous Integration (CI)?
- Continuous Integration process
- CI infrastructure
- CI tools

2

2

Integration

- Integration occurs when changes are merged with the source code repository

3

3

Broken Integration

- You have a broken integration when:
 - Source code server does not build successfully
 - Shared component works in one system, but breaks others
 - Unit tests fail
 - Code quality fails (coding conventions, quality metrics)
 - Deployment fails

4

4

Manual Integration

- Integration becomes expensive
 - if made manual (build, test, deployment, ...)
 - with too few checkin's (hours or days...)
 - if integration problems and bugs are detected too late
- Reduces desire to refactor
 - long time between integration increases risk of merge

5

5

What is Continuous Integration?

- *“Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible.”*

<http://martinfowler.com/articles/continuousIntegration.html>

6

6

Why continuous integration ?

- Maintain a code repository
- Automate the build
- Make the build self-testing
- Everyone commits to the baseline every day
- Every commit (to baseline) should be built
- Keep the build fast

7

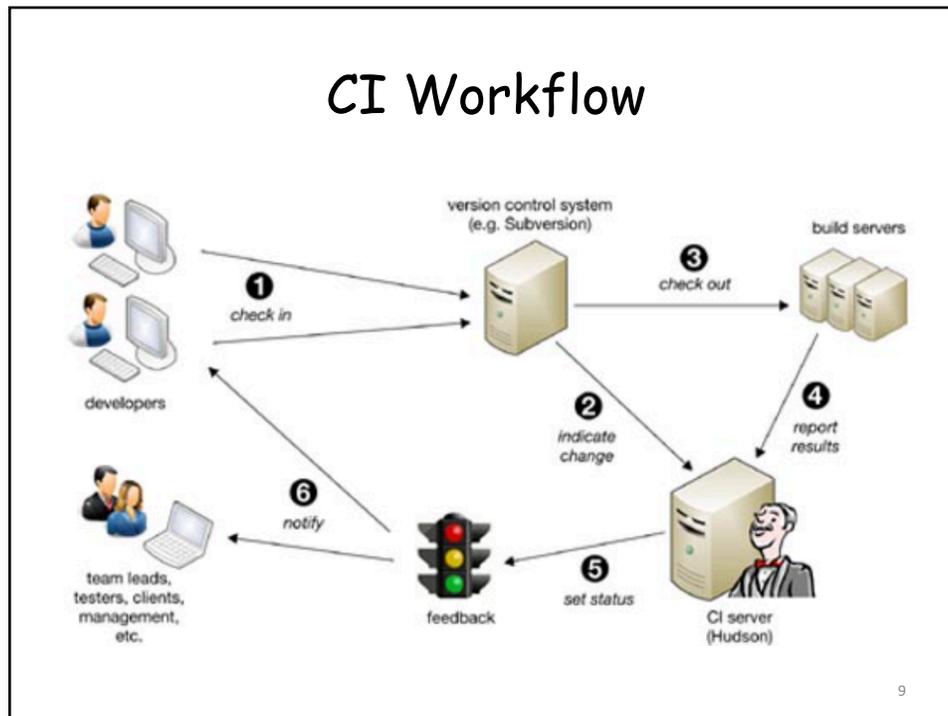
7

Why continuous integration ? (cont'd)

- Test in a clone of the production environment
- Make it easy to get the latest deliverables
- Everyone can see the results of the latest build
- Automate deployment

8

8



9

Realizing Continuous Integration

- Monitor a VCS repository for changes
 - If changes are found, then start the build
- Build your application
 - through your existing Ant or Maven scripts
- Run your xUnit Test suite
- Run code audit tools
 - Checkstyle, code coverage, ...

10

10

Realizing Continuous Integration (cont'd)

- Report on the build results
 - Send formatted email notifications
 - Publish results to a website
- (Optionally) Publish the application
- Configuration is through a central XML file

11

11

The Agile Process

- Continuous Integration is only one aspect of an overall process. For it to work best, you need to
 - Plan iteratively
 - Schedule regular releases with evolving levels of functionality
 - Implement incrementally
 - Identify and implement small work tasks
 - refactor if necessary

12

12

The Agile Process (cont'd)

- Report proactively
 - Identify exactly the contents (CIs) of any build, in both file and content
 - Automate reports!

13

13

CI Benefits

- Reduced Risks
 - Always aware of current status of the project
 - Less time spent investigating integration bugs
 - Integrating testing performed early
 - Integration bugs caught early
 - Less time wasted because of broken code in VCS
 - Broken builds caught early

14

14

CI Benefits (cont'd)

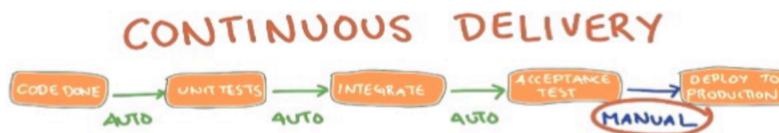
- Prove your system can build!
- Increase code quality with additional tasks

15

15

Continuous Delivery

- Continuous Delivery is the ability to get changes of all types—including new features, configuration changes, bug fixes and experiments—into production, or into the hands of users, *safely* and *quickly* in a *sustainable* way.



16

16

Continuous Deployment

- The next step of continuous delivery: Every change that passes the automated tests is deployed to production automatically. Continuous deployment should be the goal of most companies that are not constrained by regulatory or other requirements.



17

17

CI Obstacles [2]

- 1. Individuals may see CI counterproductive
 - Product managers want to launch new features
 - Project managers want the team to meet the deadline
 - Programmers want to fix the bug they are working on
 - It seems that keeping the build is an extra burden on people
 - Solution: team leaders help employees understand the costs and benefits of CI

18

18

- 2. Tough to integrate CI into an existing development flow
 - Solution
 - Give enough time for people to develop their new workflow
 - Ensure them that the company has their backs even if things might not go very smoothly at the beginning

19

19

- 3. Requiring developers of writing more test cases
 - Solution
 - Emphasize that writing test cases from early on could save a lot of time for your team and ensure high test coverage of your product
 - Embed the idea in the company culture that test cases are as valuable assets as the codebase itself.

20

20

- 4. Developers ignoring error messages
 - Developers ignore or mute the overwhelming amount of CI notifications, and may miss the updates that are relevant to them
 - Solution
 - Only send critical updates
 - Only send the notification to developers who are in charge of fixing it

21

21

- 5. Creating Fear-Driven Development
 - Being afraid of breaking the build or not passing tests could build up pressure and fear in individuals.
 - Solution
 - Make team members consider failed test as positive results, the earlier their tests fail the earlier they can resolve problems.

22

22

Reference

[1] Martin Kropp, Continuous Integration, <https://web.fhnw.ch/plattformen/swc/Literature/10-continuousintegration.pdf>. Last visited: 1/18

[2] 5 Challenges That Can Break Your Continuous Integration Efforts, <https://medium.com/flow-ci/5-challenges-that-can-break-your-continuous-integration-efforts-94aded600b3d>. Last visited: 1/18

23