Continuous Integration

Jenkins, Eclipse, JUnit

Lukas Rypl

TTC MARCONI
Czech Republic

Twitter: @LukasRypl

for BarCamp Saigon

December 11, 2011

Target Audience

- Java developers
- Programmers in any other language
- Their team leaders
- Managers
- Testers

Target Audience

- Java developers
- Programmers in any other language
- Their team leaders
- Managers
- Testers

Proffesionals who

- feel safer
- sleep better
- validate their work

Definition

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. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly.

(Martin Fowler)

Developers Work Algorithm

```
while (task.requiresWork())
{
1. Write test
2. Write implementation
}
Quick local test
Commit to VCS (cvs, svn, hg, git , ...)
```

Continuous Integration Server

- downloads source code
- builds it
- tests application
- creates package
- deploys it
- sends emails
- plays sounds
- ...

Why Continuous Integration

- run all tests properly
- produce artifacts always the same way
- blame coworkers :)
- other fun stuff

http://www.ashlux.com/wordpress/2009/07/16/psa-every-time-you-break-the-build/

Why Unit Testing

- make sure that foundations are OK
- create better (extensible) design

Tools

- Eclipse (IDE) www.eclipse.org
- SVN (VCS)
- JUnit (unit testing library) www.junit.org
- Jenkins (CI server) www.jenkins-ci.org

Demo

Books

- Kent Beck:
  Test Driven Development by Example
- Michael E. Faethers:
  Working Effectively with Legacy Code
- Martin Fowler:
  Refactoring: Improving the Design of Existing Code