



Trójmiasto JUG

INICJATYWA STUDENCKA

Gdańsk, 08.04.2014

SPOTKANIA

- 08.04 - Wprowadzenie do Agile/Scrum/Kanban
- 15.04 - Wprowadzenie do systemu budowania aplikacji Maven
- 29.04 - Wprowadzenie do testowania
- 06.05 - Specyfika testów na urządzenia mobilne
- 13.05 - I jest GIT!
- **20.05 - Wprowadzenie do Continuous Integration**
- 27.05 - User Experience / Design dla początkujących
- 03.06 - Jak programować, by nie zwariować

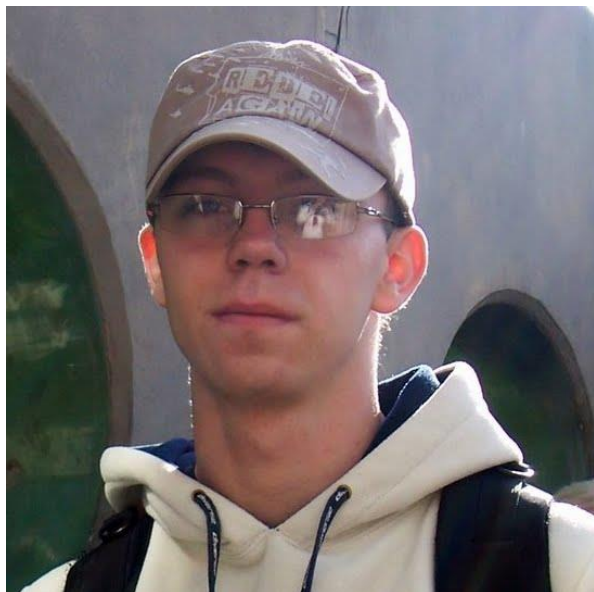


Trójmiasto JUG

WPROWADZENIE DO
CONTINUOUS INTEGRATION

Gdańsk, 20.05.2014

O MNIE



- Absolwent wydziału Fizyki Technicznej i Matematyki Stosowanej Politechniki Gdańskiej
- Projektant/programista w firmie Solwit
- Lider Trójmiasto Java User Group
- Blogger: blog.dragonia.org.pl
- Twitter: [@Smoczysko](https://twitter.com/Smoczysko)
- LinkedIn: pl.linkedin.com/pub/tukas-z-rybka/14/b78/864/

***„A man who dares to waste one hour
of time has not discovered the value
of life.”***

Charles Darwin

Kent Beck

James Grenning Robert C. Martin

Mike Beedle Jim Highsmith

Steve Mellor Arie van Bennekum

Andrew Hunt Ken Schwaber

Alistair Cockburn Ron Jeffries

Jeff Sutherland Ward Cunningham

Jon Kern Dave Thomas

Martin Fowler Brian Marick

Luty 2001, Snowbird, Utah, USA

Kent Beck

James Grenning Robert C. Martin
Mike Beedle Jim Highsmith
Steve Mellor Arie van Bennekum
Andrew Hunt Ken Schwaber
Alistair Cockburn Ron Jeffries
Jeff Sutherland Ward Cunningham
Jon Kern Dave Thomas
Martin Fowler Brian Marick

Luty 2001, Snowbird, Utah, USA

Kent Beck

„Extreme Programming Explained: Embrace Change”, 1999

Martin Fowler

„Continuous Integration”, 1 maja 2006

CZYM JEST CONTINUOUS INTEGRATION?

- Narzędzie do kompilowania i linkowania kodu źródłowego

CZYM JEST CONTINUOUS INTEGRATION?

- Narzędzie do kompilowania i linkowania kodu źródłowego
- Narzędzie do uruchamiania testów automatycznych

CZYM JEST CONTINUOUS INTEGRATION?

- Narzędzie do kompilowania i linkowania kodu źródłowego
- Narzędzie do uruchamiania testów automatycznych
- Narzędzie do wdrażania najnowszych zmian na środowiska developerskie oraz produkcyjne

CZYM JEST CONTINUOUS INTEGRATION?

- Narzędzie do kompilowania i linkowania kodu źródłowego
- Narzędzie do uruchamiania testów automatycznych
- Narzędzie do wdrażania najnowszych zmian na środowiska developerskie oraz produkcyjne
- Narzędzie do przygotowywania binarnych wersji aplikacji

CZYM JEST CONTINUOUS INTEGRATION?

- Narzędzie do kompilowania i linkowania kodu źródłowego
- Narzędzie do uruchamiania testów automatycznych
- Narzędzie do wdrażania najnowszych zmian na środowiska developerskie oraz produkcyjne
- Narzędzie do przygotowywania binarnych wersji aplikacji
- Narzędzie do (...)
- Narzędzie do (...)
- Narzędzie do (...)
-

CZYM JEST CONTINUOUS INTEGRATION?

- Narzędzie do kompilowania i linkowania kodu źródeł
- Narzędzie do uruchamiania testów automatycznych
- Narzędzie do wdrażania najnowszych zmian na środowiska developerskie i produkcyjne
- Narzędzie do przygotowania binarnych wersji aplikacji
- Narzędzie do (...)
- Narzędzie do (...)
- Narzędzie do (...)
- ...

!!!ŹŁE!!!

CZYM JEST CONTINUOUS INTEGRATION?

- Wywodzi się z Extreme Programming (XP)
- Praktyka w inżynierii programowania
- Zakłada częste łączenie (integracje) kolejnych, mniejszych zmian w kodzie z głównym branchem
- Pozwala uniknąć tzw. „merging hell”
- Zakłada się kilka integracji (merge’y) dziennie, ~ jeden commit + merge na jednego programistę w zespole



Trójmiasto JUG

WPROWADZENIE DO
CONTINUOUS INTEGRATION
(I POCHODNYCH...)

Gdańsk, 20.05.2014

JOEL TEST

1. Do you use source control?
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. **Do you use the best tools money can buy?***
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

JOEL TEST

- 1. Do you use source control?**
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?*
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

JOEL TEST

1. Do you use source control?
2. **Can you make a build in one step?**
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?*
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

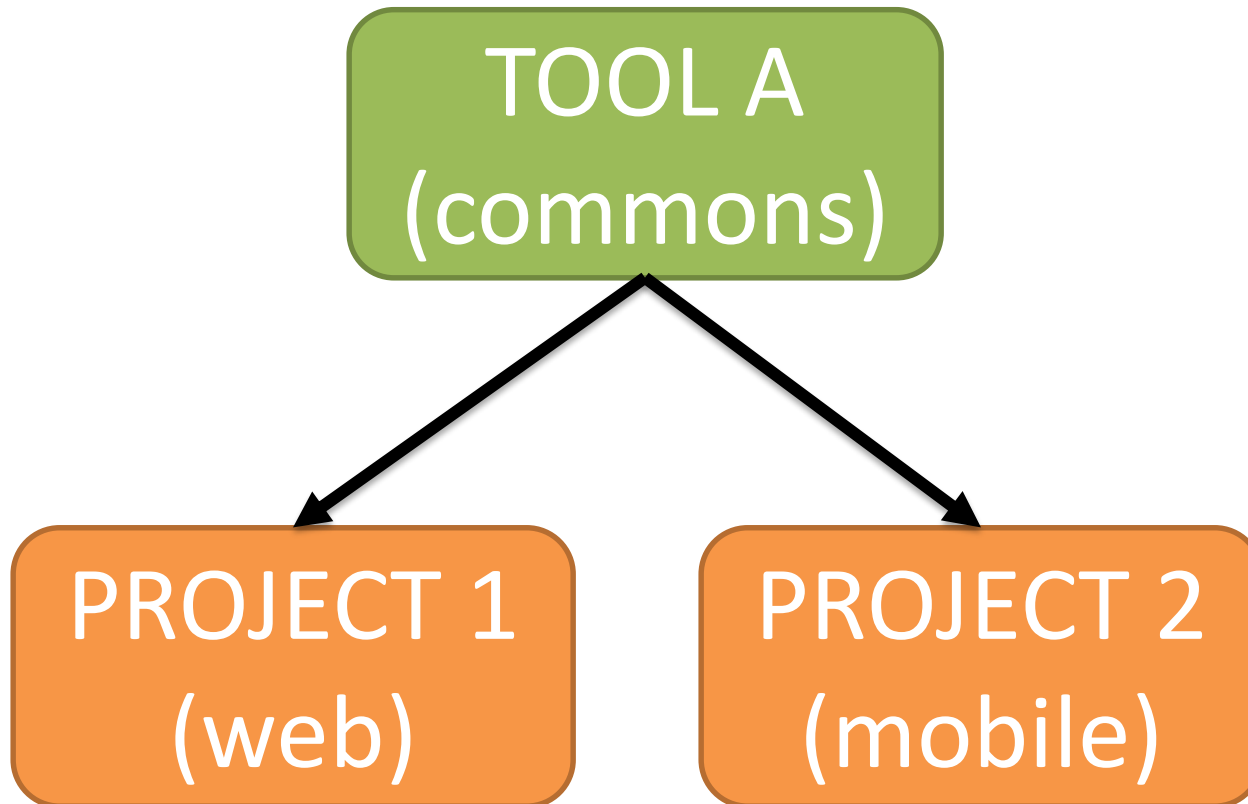
JOEL TEST

1. Do you use source control?
2. Can you make a build in one step?
- 3. Do you make daily builds?**
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?*
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

JOEL TEST

1. Do you use source control?
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
- 9. Do you use the best tools money can buy?***
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

PRZYKŁADOWY PROCES BUDOWANIA APLIKACJI



PRZYKŁADOWY PROCES BUDOWANIA APLIKACJI

1. Kompilacja TOOL A
2. Uruchomienie testów jednostkowych TOOL A
3. Przygotowanie binarnej wersji TOOL A
4. „Przekopiowanie” binarnej wersji do PROJECT 1
5. Kompilacja PROJECT 1
6. Uruchomienie testów jednostkowych PROJECT 1
7. Wygenerowanie binarnej wersji PROJECT 1
8. „Przekopiowanie” binarnej wersji do PROJECT 2
9. Kompilacja PROJECT 2
10. Uruchomienie testów jednostkowych PROJECT 2
11. Wygenerowanie binarnej wersji PROJECT 2

PRZYKŁADOWY PROCES BUDOWANIA APLIKACJI

1. Kompilacja TOOL A
2. Uruchomienie testów jednostkowych TOOL A
3. Przygotowanie binarnej wersji TOOL A

FAZA 1

1. „Przekopiowanie” binarnej wersji do PROJECT 1
2. Kompilacja PROJECT 1
3. Uruchomienie testów jednostkowych PROJECT 1
4. Wygenerowanie binarnej wersji PROJECT 1

FAZA 2

1. „Przekopiowanie” binarnej wersji do PROJECT 2
2. Kompilacja PROJECT 2
3. Uruchomienie testów jednostkowych PROJECT 2
4. Wygenerowanie binarnej wersji PROJECT 2

FAZA 3

CONTINUOUS INTEGRATION TOOLS (SERVERS)



„(...) I had this great idea of a web framework, and I wanted an excuse to actually implement it. This was around the time when everyone had his own framework, so I suppose I couldn't help myself.”

Kohsuke Kawaguchi

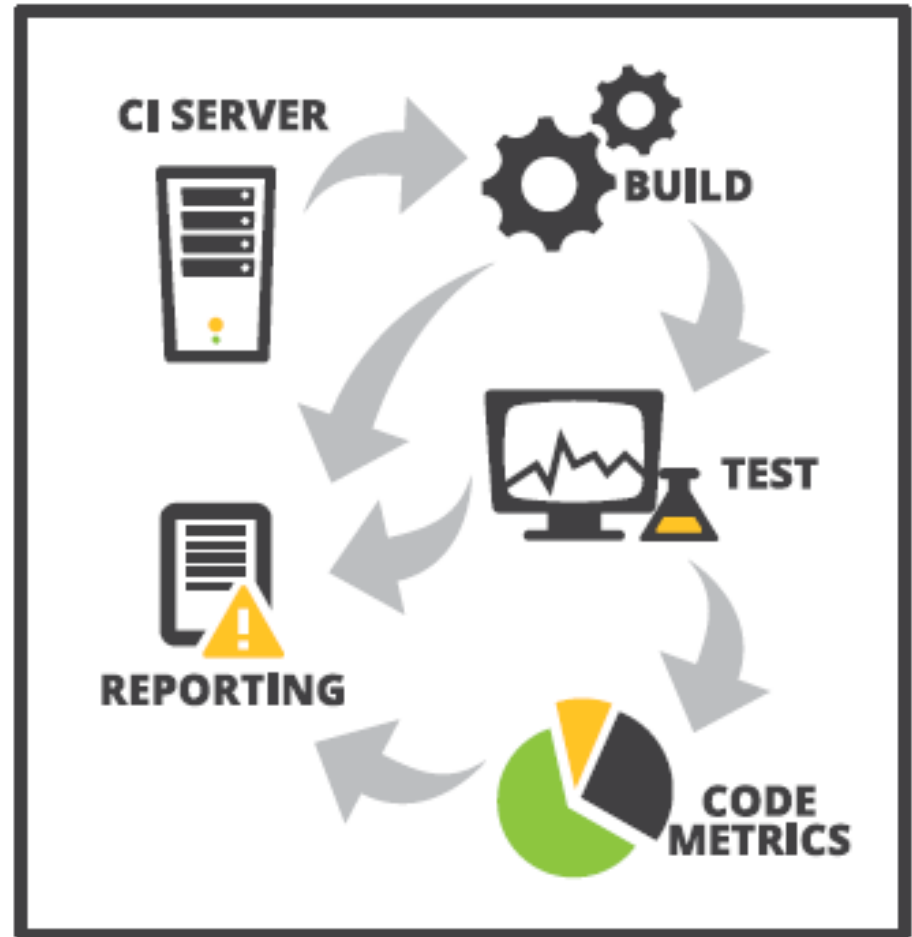
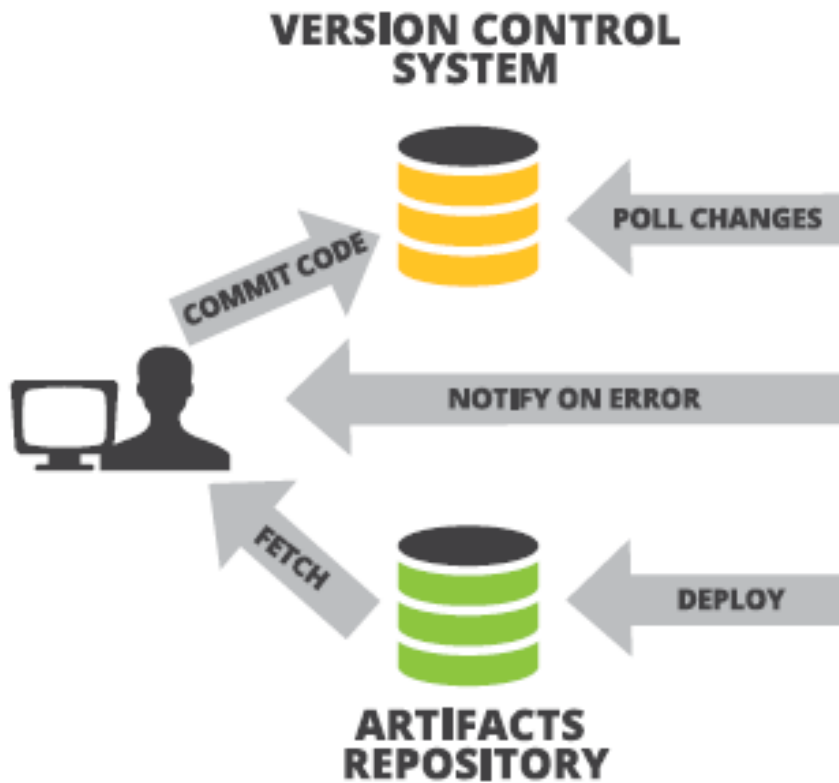
JENKINS - HISTORIA

- 2004: rozpoczęcie prac przez Kawaguchi'ego nad projektem Hudson
- 2005: Hudson 1.0
- 2007: powstaje pierwszych 6 rozszerzeń do Hudsona
- 2008: ilość rozszerzeń rośnie do 27
- 2009: ilość rozszerzeń rośnie do 94
- 2010: Oracle przejmuje Sun Microsystems
- **2011: Kawaguchi odchodzi z Oracle, powstaje projekt Jenkins**
- 2012: 469 rozszerzeń, ponad 30tys aktywnych instalacji Jenkinsa
- 2013: 636 rozszerzeń, ponad 50tys aktywnych instalacji

JENKINS – PODSTAWOWE CECHY

- Łatwa instalacja (plik war lub natywne paczki dla 9 systemów operacyjnych, w tym Windows, Linux oraz Max OS X)
- Łatwe i częste aktualizacje (release'y częściej niż raz w tygodniu)
- Ogromna baza pluginów (tworzonych przez społeczność)
- Prostota obsługi
- Możliwość łączenia wielu serwerów w „sieć” (master/slave) i automatyczne rozkładanie obciążenia
- Nie zależny od platformy i języka programowania
- Łatwość skryptowania

PRZYKŁADOWY PROCES BUDOWANIA APLIKACJI



PRZYKŁADOWY BUILD JENKINSA - KONFIGURACJA

Item name

Build a free-style software project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this ca

Build a maven2/3 project

Build a maven 2/3 project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Build multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platfor

Monitor an external job

This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is de
system. See [the documentation for more details](#).

Copy existing Item

Copy from

OK

PRZYKŁADOWY BUILD JENKINSA - KONFIGURACJA

Source Code Management

- CVS
- CVS Projectset
- File System
- Git

Repositories

Repository URL

Credentials

Branches to build

Branch Specifier (blank for 'any')

Repository browser

PRZYKŁADOWY BUILD JENKINSA - KONFIGURACJA

Build Triggers

- Build whenever a SNAPSHOT dependency is built
- Trigger builds remotely (e.g., from scripts)
- Build after other projects are built
- Build periodically

Schedule

```
H/5 * * * *
```

Would last have run at wtorek, 20 maj 2014 00:46:35 CEST; would next run at wtorek, 20 maj 2014 00:51:35 CEST.

- Poll SCM

PRZYKŁADOWY BUILD JENKINSA - KONFIGURACJA

Build

Root POM

`pom.xml`


Goals and options



`package`

PRZYKŁADOWY BUILD JENKINSA - URUCHOMIENIE

Build History (trend) —

#1	May 20, 2014 12:51:58 AM
----	--



 [RSS for all](#)  [RSS for failures](#)

PRZYKŁADOWY BUILD JENKINSA - URUCHOMIENIE

```
Building in workspace /var/lib/jenkins/jobs/jenkins-maven-example/workspace
Cloning the remote Git repository
Cloning repository https://github.com/trojmiasto-jug/jenkins-maven-example.git
git --version
git version 1.7.5.4
Checking out Revision 83eb7b7d77903910ecdd98d3eb8cd173991646ae (origin/master)

Executing Maven: -B -f /var/lib/jenkins/jobs/jenkins-maven-example/workspace/pom.xml package
[INFO] Scanning for projects...
[HUDSON] Collecting dependencies info
[INFO]
[INFO] -----
[INFO] Building jenkins-maven-example 1.0-SNAPSHOT
[INFO] -----
```

PRZYKŁADOWY BUILD JENKINSA - URUCHOMIENIE

```
-----  
T E S T S  
-----
```

```
Running pl.jug.trojmiasto.AppTest
```

```
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.015 sec
```

```
Results :
```

```
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
```

```
[INFO] Building jar: /var/lib/jenkins/jobs/jenkins-maven-example/workspace/target/jenkins-maven-example-1.0-SNAPSHOT.jar
```

```
[INFO] -----
```

```
[INFO] BUILD SUCCESS
```

```
[INFO] -----
```

```
[INFO] Total time: 19.027s
```

```
[INFO] Finished at: Tue May 20 00:52:31 CEST 2014
```

```
[INFO] Final Memory: 16M/66M
```

```
[INFO] -----
```

```
Finished: SUCCESS
```

JENKINS - CUSTOMIZACJA

S	W	Name ↓	Last Success	Last Failure	Last Duration	
		jenkins-maven-example	30 sec - #2	N/A	12 sec	

PRZYKŁADOWY BUILD JENKINSA – POPSUTE TESTY

```
-----  
T E S T S  
-----  
Running pl.jug.trojmiasto.AppTest  
Tests run: 1, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.025 sec <<< FAILURE!  
  
Results :  
  
Failed tests:   testApp(pl.jug.trojmiasto.AppTest)  
  
Tests run: 1, Failures: 1, Errors: 0, Skipped: 0  
  
[ERROR] There are test failures.
```

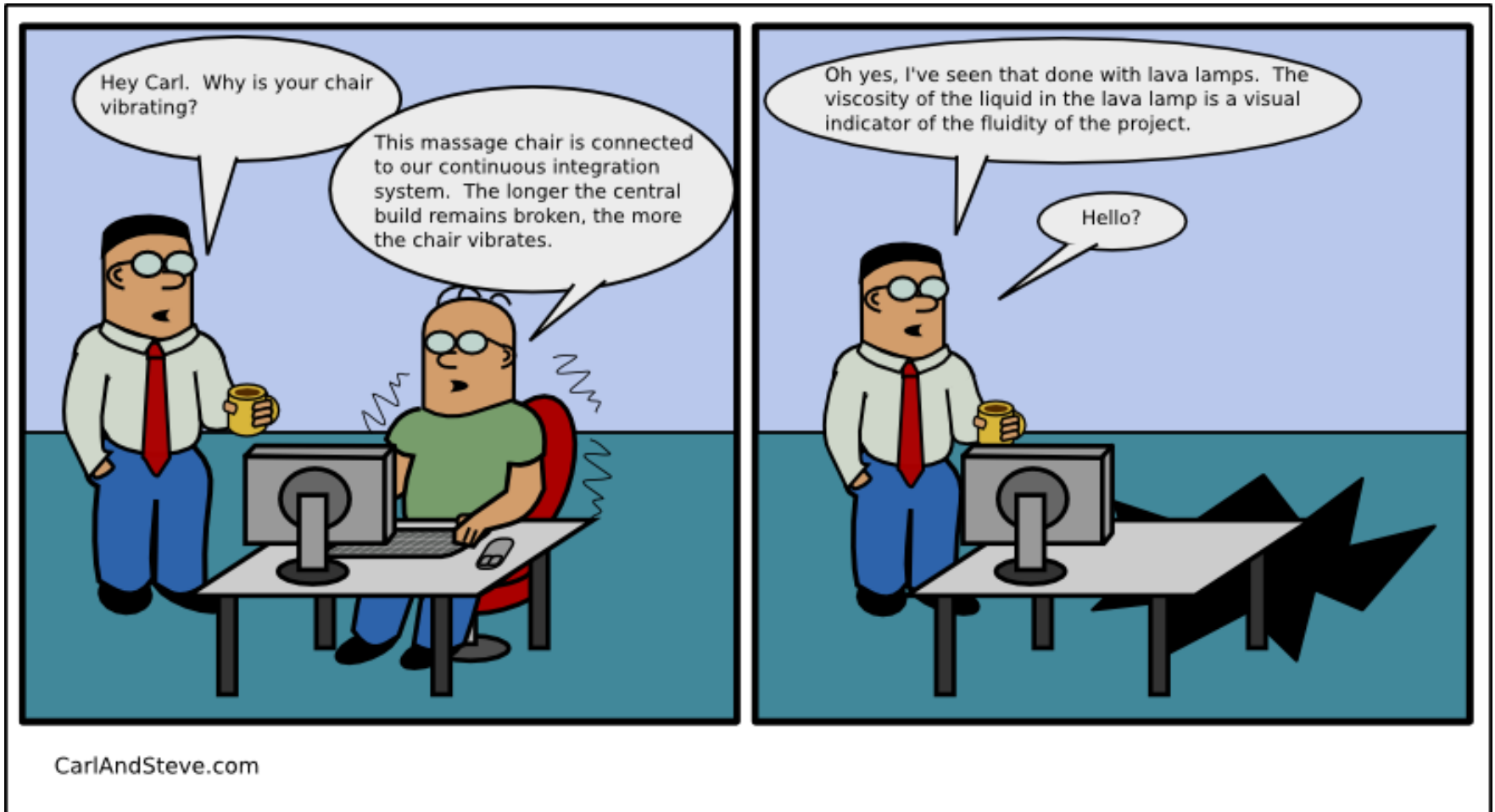
PRZYKŁADOWY BUILD JENKINSA – POPSUTE TESTY

```
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 6.069s  
[INFO] Finished at: Tue May 20 01:14:56 CEST 2014  
[INFO] Final Memory: 16M/136M  
[INFO] -----
```

Finished: UNSTABLE

S	W	Name ↓	Last Success
		jenkins-maven-example	4 min 53 sec - #5
		jenkins-maven-example-fail	1 min 10 sec - #1

ZDROWY ROZSADEK



PYTANIA?

Q&A

REFERENCJE

- „The Joel Test: 12 Steps to Better Code”, Joel Spolsky, <http://www.joelonsoftware.com/articles/fog0000000043.html>
- „Daily Builds Are Your Friend”, Joel Spolsky, <http://www.joelonsoftware.com/articles/fog0000000043.html>
- „Why Devs <3 CI: A Guide to Loving Continuous Integration”, ZeroTurnaround, RebelLabs, <http://zeroturnaround.com/rebellabs/rebellabs-report-why-devs-love-ci-a-guide-to-loving-continuous-integration/>
- „Jenkins CI: The Origins of Butler, Build Masters and Bowties”, ZeroTurnaround, RebelLabs, <http://zeroturnaround.com/rebellabs/rebellabs-report-jenkins-ci-the-origins-of-butlers-build-masters-and-bowties/>

DZIĘKUJĘ!