



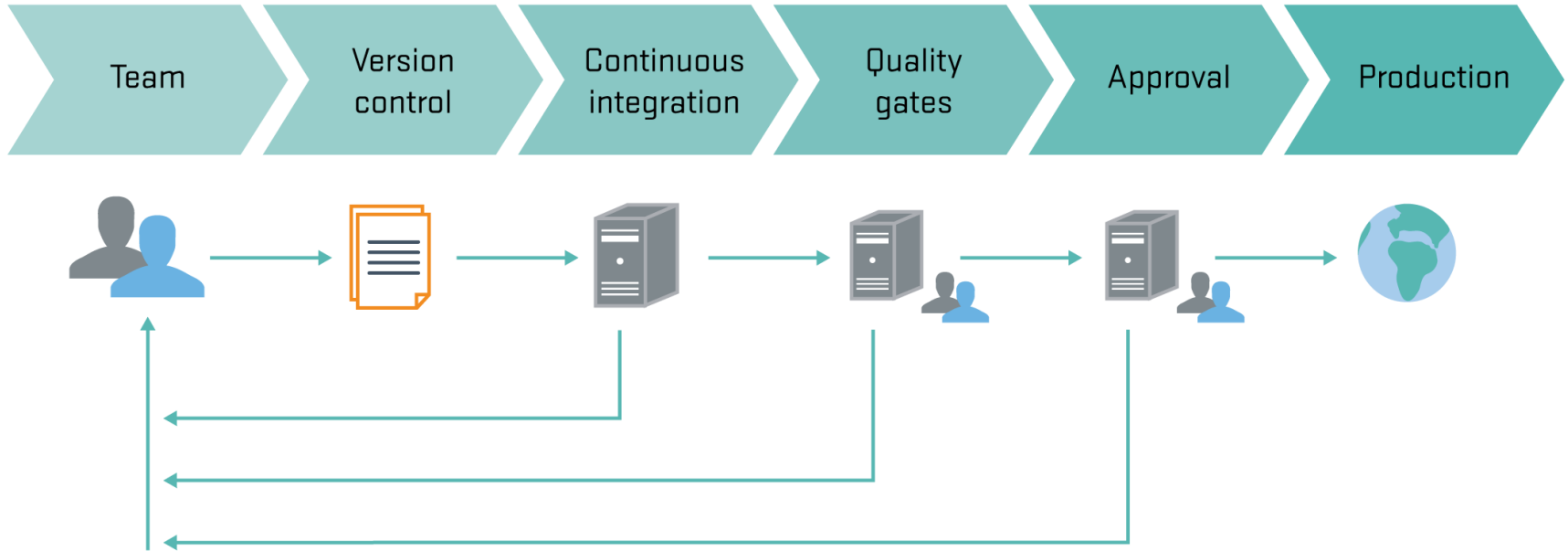
# Continuous Delivery

by Vlad Ungreanu

## Agenda

### Continuous Delivery

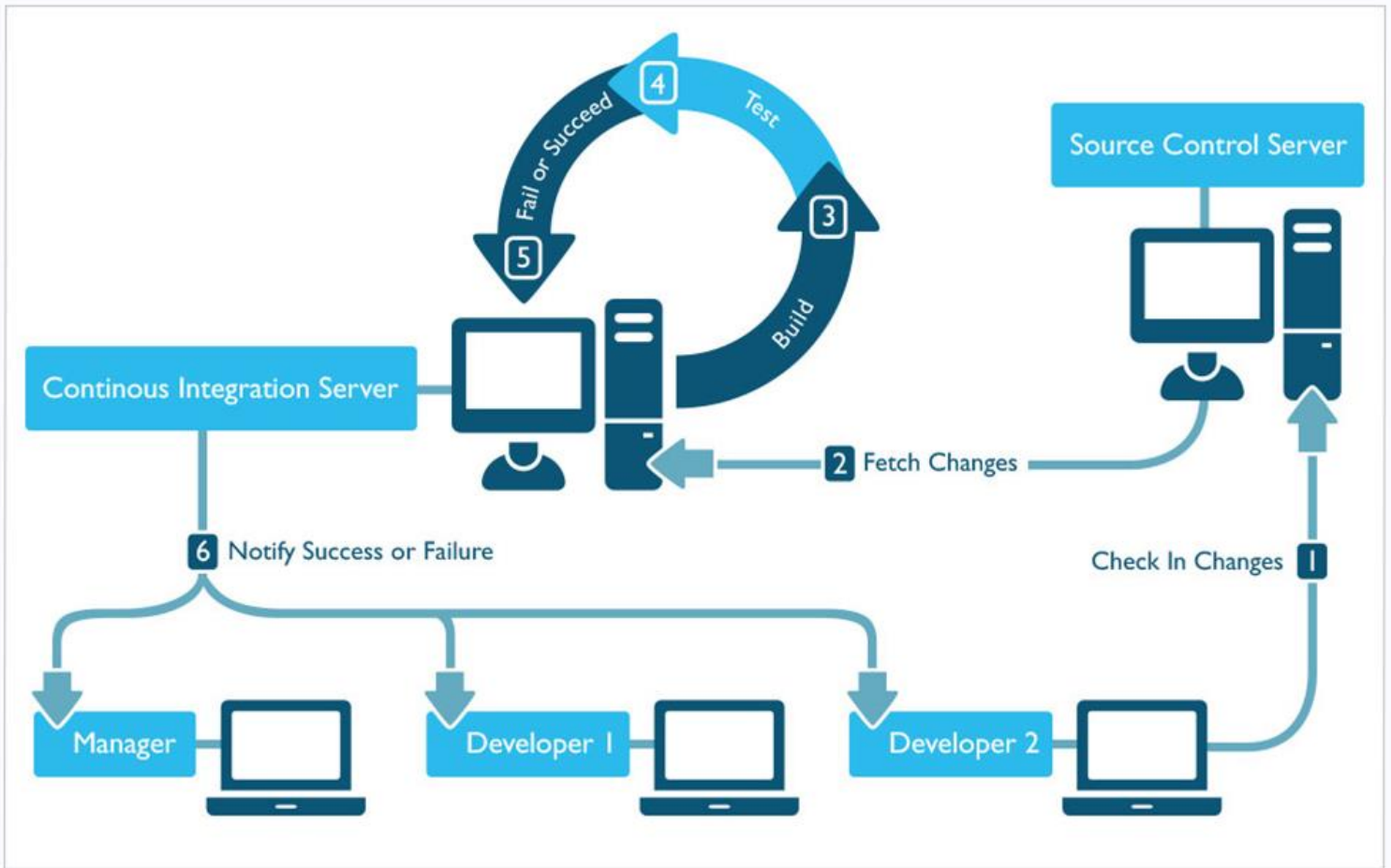
- Development Pipeline
- Target Results of Continuous Delivery
- Benefits of Continuous Delivery
- Continuous Delivery Feedback Process
- Continuous Integration
- Continuous Integration Components
- Continuous Integration Tools
- Git
- Jenkins
- Sonar
- Continuous Integration Practices



- An *automated implementation* of an application build, deploy, test and release process
- Any change to the application (code, tests, configuration) will trigger a new instance of the development pipeline
- Code is compiled and the application build, the tests are run and if everything is in order then the application can be released
- Usually after the application is build and the unit tests are run, there are several other types of automated tests run like “*Acceptance Tests*”, “*Capacity Tests*” and “*Manual Tests*” before an application can be released

- Fast delivery for each version of an application
- Reliable quality assurance for developers, testers and customer
- Empower Agile and SCRUM teams
- Facilitate feedback and collaboration with all participating disciplines within the company, and with any client side involved entities

- Avoid human errors and unintentional skipping of certain steps
- Ability to track and discover bugs or problems in a system that is not transparent
- Simplify the build, test and release process making it reusable, reliable and (obviously) automated
- Automation of an application delivery process encourages inter-disciplinary collaboration within the company
- Automaton is reliable, controllable, configurable, cheap, predictable and easily repeatable



- Any change in the application can change the application behavior
- Any change should lead to build and test
- Anything that differs from one environment to another is consider configuration
- Environment changes should trigger testing
- If data changes the application should still work as expected
- Data changes should trigger testing
- Results:
  - Code build => code syntax is valid
  - Unit testing => application behavior is as expected
  - Test coverage => quality assurance criteria (Sonar)
  - Acceptance tests => application respect business criteria
  - NF tests => application is stable and secure

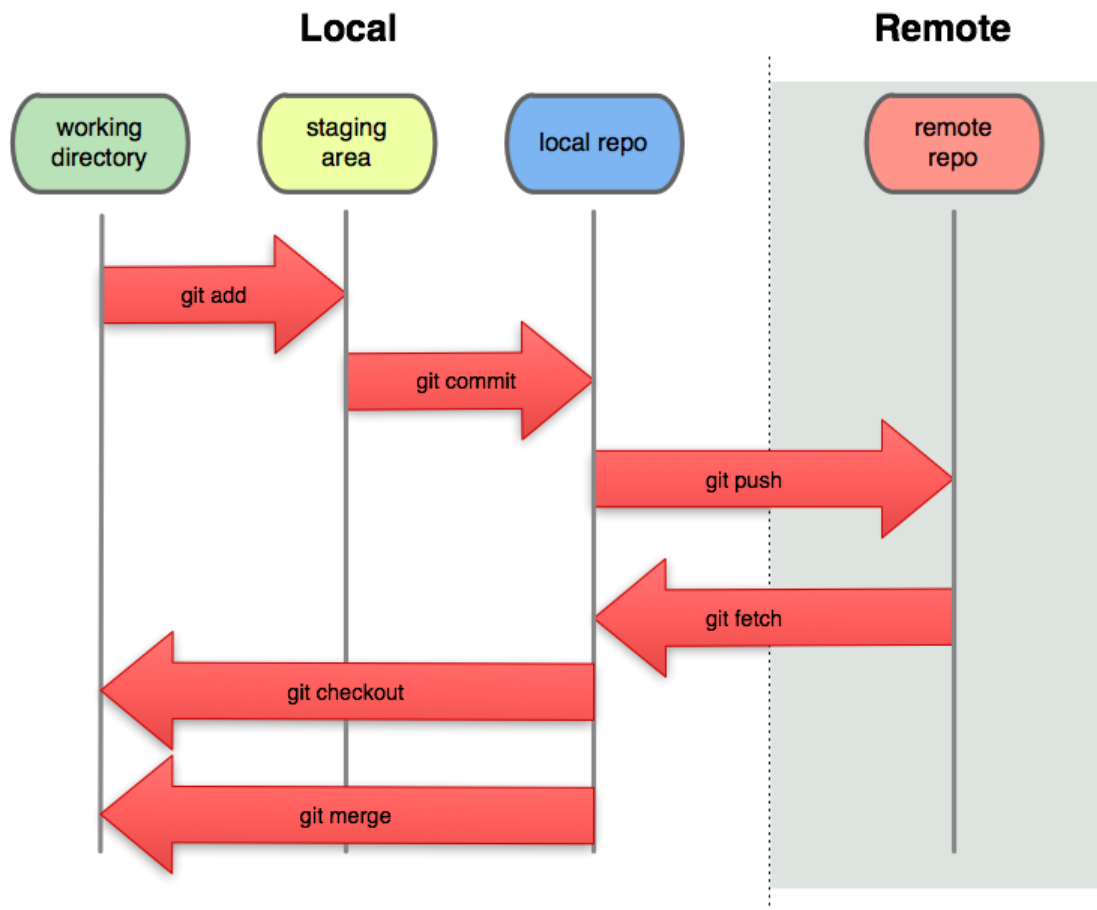


- In a broad sense [Continuous Integration](#) refers to the process of automatically triggering build and tests after each change to the application
- In a strict sense Continuous Integration refers to the process of monitoring the source control repository where the code is kept and triggering a build and test routine every time a change is committed

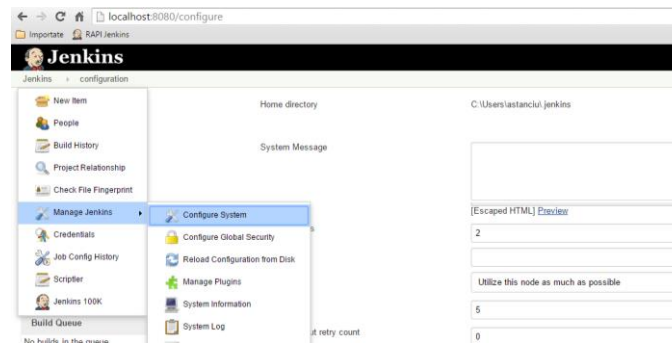
# Continuous Integration components



- Source Control
  - Git
  - SVN
- Build
  - Maven plugins
  - Gradle plugins
  - Ant plugins
- Build/ test/ deploy jobs
  - Jenkins
  - Bamboo
  - Go
- Deployment Server
  - Similar (if not identical) to the production environment



- Install
- <https://jenkins-ci.org/>
- Prerequisites: Java JDK > 1.5, a servlet container - Jetty (included) or another one(e.g. Tomcat)
  
- Start
- - through the built in Jetty servlet container:
- `$ java -jar jenkins.war`
  
- `http://myServer:8080`
  
- Configure



- Create a Job Item – Jenkins > New Item    Jenkins > New\_Job > Configure

#### Source Code Management

- None
- CVS
- CVS Projectset
- Git

Repositories

Repository URL

Credentials

 Add

#### Build Triggers

- Build whenever a SNAPSHOT dependency is built
- Build after other projects are built
- Build periodically

Schedule

Would last have run at Monday, March 30, 2015 9:16:53 AM FET; would next run at Monday, March 30, 2015 10:16:53 AM FET.

- Poll SCM

Schedule

 Do you really mean "every minute" when you say "\* \* \* \* \*"? Perhaps you meant "H \* \* \* \* \*" to poll once per hour



MINUTES Minutes in one hour (0-59)

HOURS Hours in one day (0-23)

DAYMONTH Day in a month (1-31)

MONTH Month in a year (1-12)

DAYWEEK Day of the week (0-7)

- 0 and 7 are Sunday

`*/5 * * * *` or `H/5 * * * *` - every 5 minutes or `0 8 * * * *` - every day at 8h00

`H` (for “hash”) parameter is used to allow periodically scheduled tasks to produce even load on the system

`0 0 * * *` for a dozen daily jobs will cause a large spike at midnight.

`H H * * *` would still execute each job once a day, but not all at the same time, better using limited resources.


The `H` symbol can be thought of as a random value over a range, but it actually is a hash of the job name, not a random function, so that the value remains stable for any given project.

- Build phase

**Build**

|                   |                                                                |
|-------------------|----------------------------------------------------------------|
| Root POM          | <input type="text" value="wp-cr-app/pom.xml"/>                 |
| Goals and options | <input type="text" value="clean install -Pcoverage-per-test"/> |

- Sonar configuration

 **Invoke Standalone Sonar Analysis**

|                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Task to run                | <input type="text"/>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| JDK                        | <input type="text" value="(Inherit From Job)"/><br><small>JDK to be used for this sonar analysis</small>                                                                                                                                                                                                                                                                                                                                                                                             |
| Path to project properties | <input type="text"/>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Project properties         | <input type="text" value="sonar.projectKey=wp:cr&lt;br/&gt;sonar.projectName=WP-CR&lt;br/&gt;sonar.projectVersion=1.0.0&lt;br/&gt;sonar.sources=wp-cr-app/src/main/java&lt;br/&gt;sonar.tests=wp-cr-app/src/test/java&lt;br/&gt;sonar.binaries=wp-cr-app/target/classes&lt;br/&gt;sonar.junit.reportsPath=wp-cr-app/target/surefire-reports&lt;br/&gt;sonar.jacoco.reportPath=wp-cr-app/target/jacoco.exec&lt;br/&gt;sonar.exclusions=wp-cr-app/src/main/java/com/worldpay/cr/**/domain/**/*.java"/> |
| JVM Options                | <input type="text"/>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |



Version 1.0.0 - Feb 24 2015 17:15 Δ over 30 days (Jan 26 2015)

|                                                                                                                                     |                                              |                                                    |                                                      |                                                     |                                                                 |                                                   |                                  |                       |                        |                      |                      |                    |
|-------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|----------------------------------------------------|------------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------------------|---------------------------------------------------|----------------------------------|-----------------------|------------------------|----------------------|----------------------|--------------------|
| <b>Lines Of Code</b><br><b>4,256</b> (+2,414)                                                                                       | <b>Files</b><br><b>86</b> (+31)              | <b>Functions</b><br><b>407</b> (+220)              | <b>SQALE Rating</b><br><b>A</b>                      | <b>Technical Debt Ratio</b><br><b>0.1%</b> (-3.8)   |                                                                 |                                                   |                                  |                       |                        |                      |                      |                    |
| <b>Java</b>                                                                                                                         | <b>Directories</b><br><b>53</b> (+22)        | <b>Lines</b><br><b>5,781</b> (+3,141)              | <b>Classes</b><br><b>86</b> (+30)                    | <b>Statements</b><br><b>1,226</b> (+725)            | <b>Accessors</b><br><b>31</b> (-52)                             | <b>Technical Debt</b><br><b>1h 15min</b> (-4d 2h) | <b>Issues</b><br><b>6</b> (-151) | <b>Blocker</b> 0 (-2) | <b>Critical</b> 0 (-2) | <b>Major</b> 2 (-93) | <b>Minor</b> 3 (-52) | <b>Info</b> 1 (-2) |
| <b>Duplications</b><br><b>0.0%</b> (-13.4)                                                                                          | <b>Lines</b><br><b>0</b> (-354)              | <b>Blocks</b><br><b>0</b> (-6)                     | <b>Files</b><br><b>0</b> (-2)                        | <b>Directory Tangle Index</b><br><b>1.5%</b> (-5.9) | <b>Dependencies To Cut</b><br>Between Directories <b>1</b> (+0) | Between Files <b>1</b> (-1)                       |                                  |                       |                        |                      |                      |                    |
| <b>Complexity</b><br><b>1.3</b> (+0.0) /function<br><b>6.2</b> (+1.9) /class<br><b>6.2</b> (+1.8) /file<br>Total: <b>530</b> (+290) |                                              | <b>Unit Tests Coverage</b><br><b>96.2%</b> (+70.9) | <b>Unit Test Success</b><br><b>100.0%</b> (+0.0)     |                                                     |                                                                 |                                                   |                                  |                       |                        |                      |                      |                    |
| <b>Events</b> <span>All</span>                                                                                                      | <b>Line Coverage</b><br><b>96.1%</b> (+70.4) | <b>Condition Coverage</b><br><b>97.4%</b> (+77.0)  | <b>Failures</b> <b>0</b> (+0)                        | <b>Errors</b> <b>0</b> (+0)                         | <b>Tests</b> <b>206</b> (+173)                                  | <b>Skipped</b> <b>0</b> (+0)                      |                                  |                       |                        |                      |                      |                    |
| Feb 24 2015                                                                                                                         | Version                                      | 1.0.0                                              | <b>Execution Time</b><br><b>22.3 sec</b> (+17.6 sec) |                                                     |                                                                 |                                                   |                                  |                       |                        |                      |                      |                    |

- Submit code changes frequently, because it limits the impact the changes have on the build and tests.
- Build comprehensive tests to cover the code as much as possible. Without tests the continuous integration only reflects the absence of compilation errors.
- Tests must not take too long. The acceptable duration is around 5 minutes.
- Run tests locally before submitting code changes.

**THANK YOU!**

**Vlad Costel Ungureanu**  
[ungureanu\\_vlad\\_costel@yahoo.com](mailto:ungureanu_vlad_costel@yahoo.com)

**This is a free course from [LearnStuff.io](https://learnstuff.io)  
– not for commercial use –**