

# Jenkins Continuous Build System

Jesse Bowes

CSCI-5828

Spring 2012

# Executive summary

- Continuous integration systems are a vital part of any Agile team because they help enforce the ideals of Agile development
- Jenkins, a continuous build tool, enables teams to focus on their work by automating the build, artifact management, and deployment processes
- Jenkins' core functionality and flexibility allow it to fit in a variety of environments and can help streamline the development process for all stakeholders involved

# Agenda

- Continuous Integration (CI)
  - What is it?
  - What are the benefits?
  - Continuous Build Systems
- Jenkins
  - What is it?
  - Where does it fit in?
  - Why should I use it?
  - What can it do?
  - How does it work?
  - Where is it used?
  - How can I get started?
- Putting it all together
- Conclusion
- References

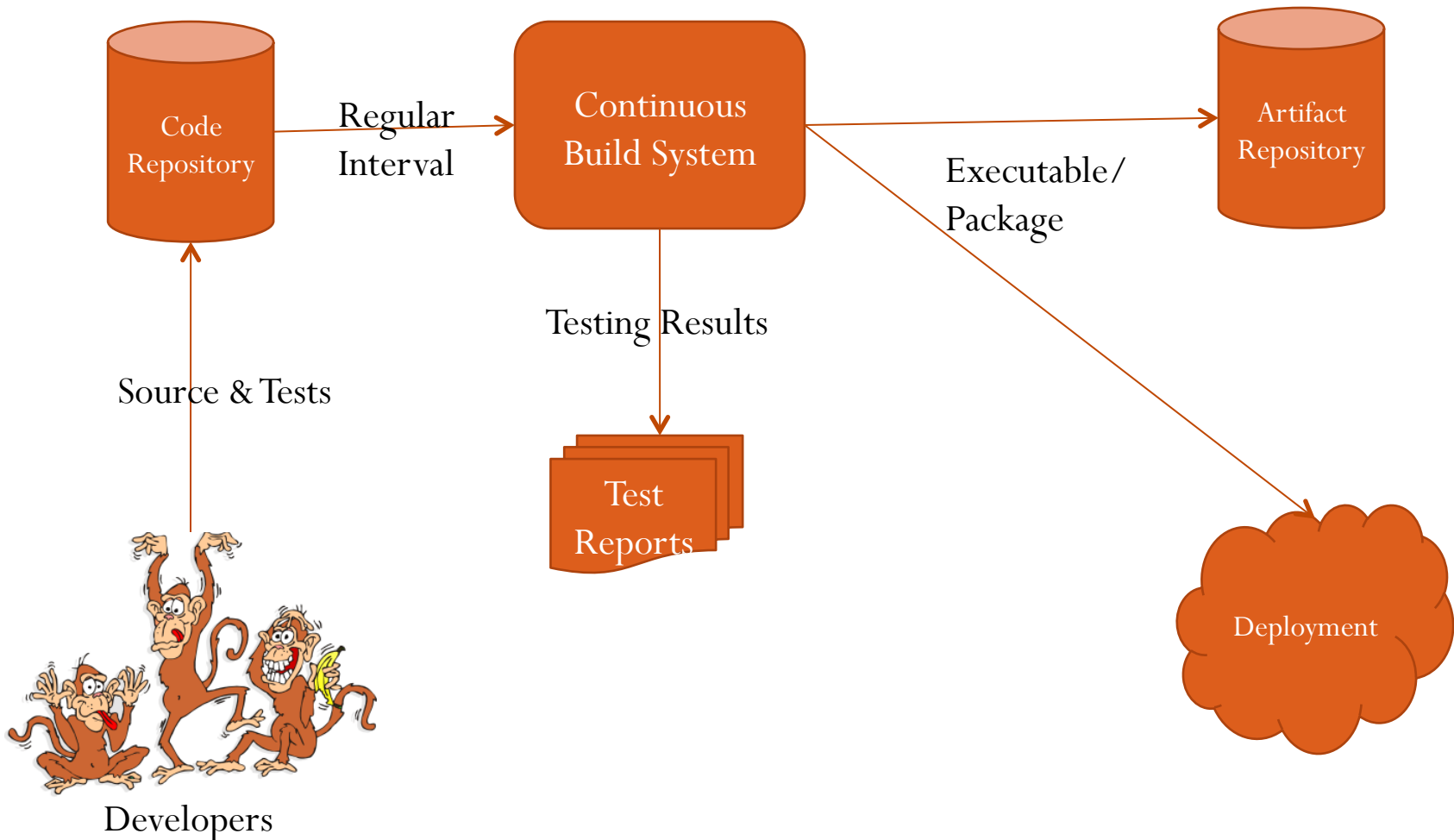
# CI - Defined

- “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” – Martin Fowler

# CI – What does it really mean?

- At a regular frequency (ideally at every commit), the system is:
  - Integrated
    - All changes up until that point are combined into the project
  - Built
    - The code is compiled into an executable or package
  - Tested
    - Automated test suites are run
  - Archived
    - Versioned and stored so it can be distributed as is, if desired
  - Deployed
    - Loaded onto a system where the developers can interact with it

# CI - Workflow



# CI – Benefits

- Immediate bug detection
- No integration step in the lifecycle
- A deployable system at any given point
- Record of evolution of the project

# CI – The tools

- Code Repositories
  - SVN, Mercurial, Git
- Continuous Build Systems
  - **Jenkins**, Bamboo, Cruise Control
- Test Frameworks
  - JUnit, Cucumber, CppUnit
- Artifact Repositories
  - Nexus, Artifactory, Archiva



# Jenkins

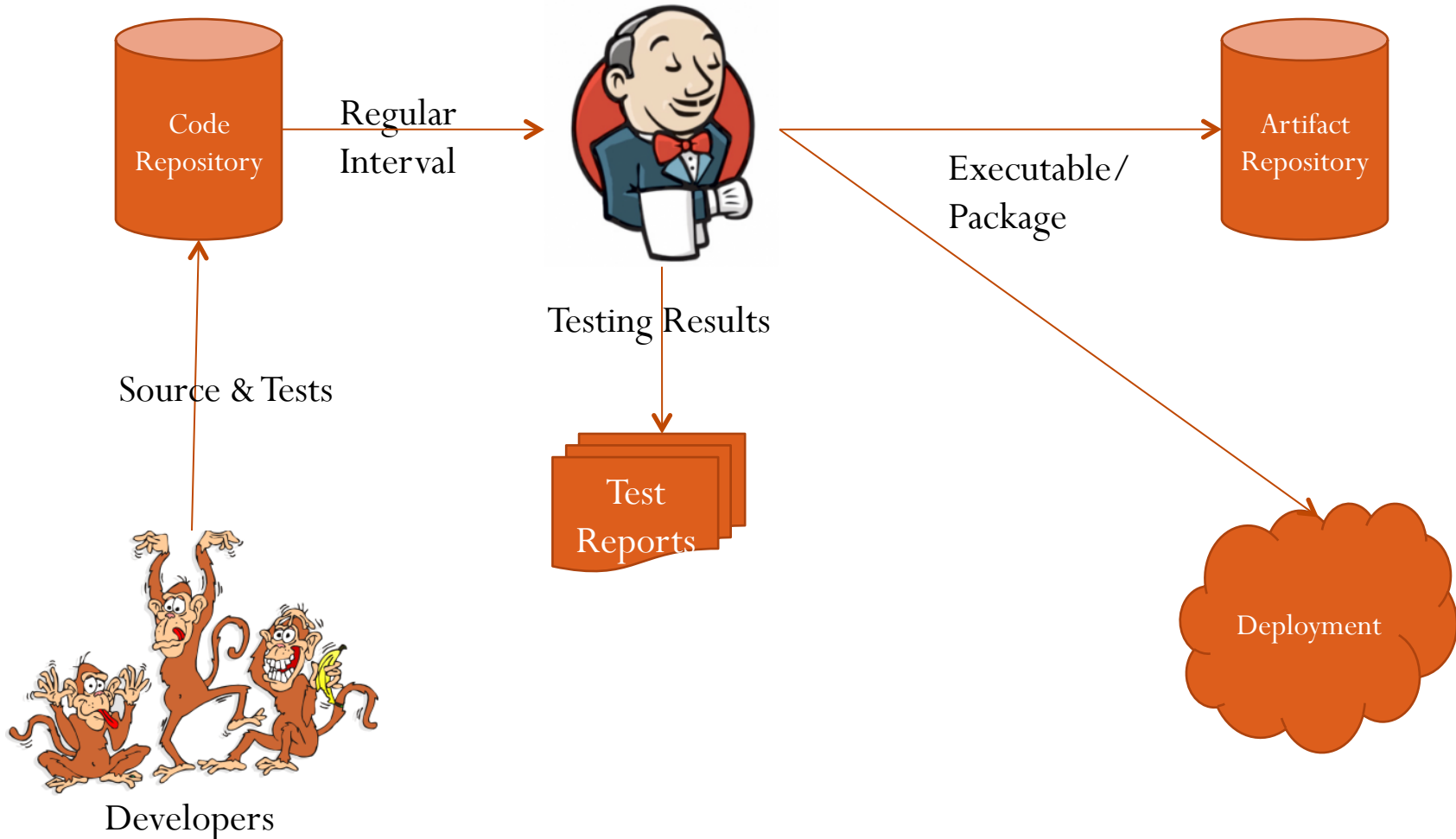


- Branched from Hudson
- Java based Continuous Build System
- Runs in servlet container
  - Glassfish, Tomcat
- Supported by over 400 plugins
  - SCM, Testing, Notifications, Reporting, Artifact Saving, Triggers, External Integration
- Under development since 2005
- <http://jenkins-ci.org/>

# Jenkins - History

- 2005 - Hudson was first release by Kohsuke Kawaguchi of Sun Microsystems
- 2010 – Oracle bought Sun Microsystems
  - Due to a naming dispute, Hudson was renamed to Jenkins
  - Oracle continued development of Hudson (as a branch of the original)

# Jenkins – Fitting in



# Why Jenkins? Flexibility!

- Jenkins is a highly configurable system by itself
- The additional community developed plugins provide even more flexibility
- By combining Jenkins with Ant, Gradle, or other Build Automation tools, the possibilities are limitless

# Why Jenkins? Award winning!

- InfoWorld Bossies Award, 2011



- O'Reilly Open-Source Award, 2011



- ALM&SCM, SDTimes 100, 2010, 2011



- GlassFish Community Innovation Award 2008



- Duke's Choice Award 2008



# Why Jenkins? Free/OSS

- Jenkins is released under the MIT License
- There is a large support community and thorough documentation
- It's easy to write plugins
- Think something is wrong with it? You can fix it!

# What can Jenkins do?

- Generate test reports
- Integrate with many different Version Control Systems
- Push to various artifact repositories
- Deploys directly to production or test environments
- Notify stakeholders of build status
- ...and much more

# How Jenkins works - Setup

- When setting up a project in Jenkins, out of the box you have the following general options:
  - Associating with a version control server
  - Triggering builds
    - Polling, Periodic, Building based on other projects
  - Execution of shell scripts, bash scripts, Ant targets, and Maven targets
  - Artifact archival
  - Publish JUnit test results and Javadocs
  - Email notifications
- As stated earlier, plugins expand the functionality even further



# How Jenkins works - Building

- Once a project is successfully created in Jenkins, all future builds are automatic
- Building
  - Jenkins executes the build in an executor
    - By default, Jenkins gives one executor per core on the build server
  - Jenkins also has the concept of slave build servers
    - Useful for building on different architectures
    - Distribution of load

# How Jenkins works - Reporting

- Jenkins comes with basic reporting features
  - Keeping track of build status
    - Last success and failure
    - “Weather” – Build trend
- These can be greatly enhanced with the use of pre-build plugins
  - Unit test coverage
  - Test result trending
  - Findbugs, Checkstyle, PMD

# Jenkins by example – Main Page

The screenshot shows the Jenkins main page. At the top is a blue header with the Jenkins logo, a search bar, and a 'log in' link. Below the header, there are links for 'People' and 'Build History'. The 'Build Queue' section shows 'No builds in the queue.' The 'Build Executor Status' section shows two executors, both 'Idle'. The 'Continuous Integration Jobs' section shows a table of jobs.

**Jenkins**  [log in](#) [ENABLE AUTO REFRESH](#)

[People](#) [Build History](#)

**Build Queue**  
No builds in the queue.

**Build Executor Status**

#	Status
1	Idle
2	Idle

**Continuous Integration Jobs:**

All

S	W	Job ↓	Last Success	Last Failure	Last Duration
		<a href="#">Spago4Q</a>	9 hr 32 min ( <a href="#">#416</a> )	3 days 9 hr ( <a href="#">#413</a> )	N/A
		<a href="#">Spago4Q-Build</a>	9 hr 30 min ( <a href="#">#830</a> )	N/A	N/A
		<a href="#">SpagoBIProject</a>	6 mo 29 days ( <a href="#">#2</a> )	N/A	N/A

Icon: [S](#) [M](#) [L](#)

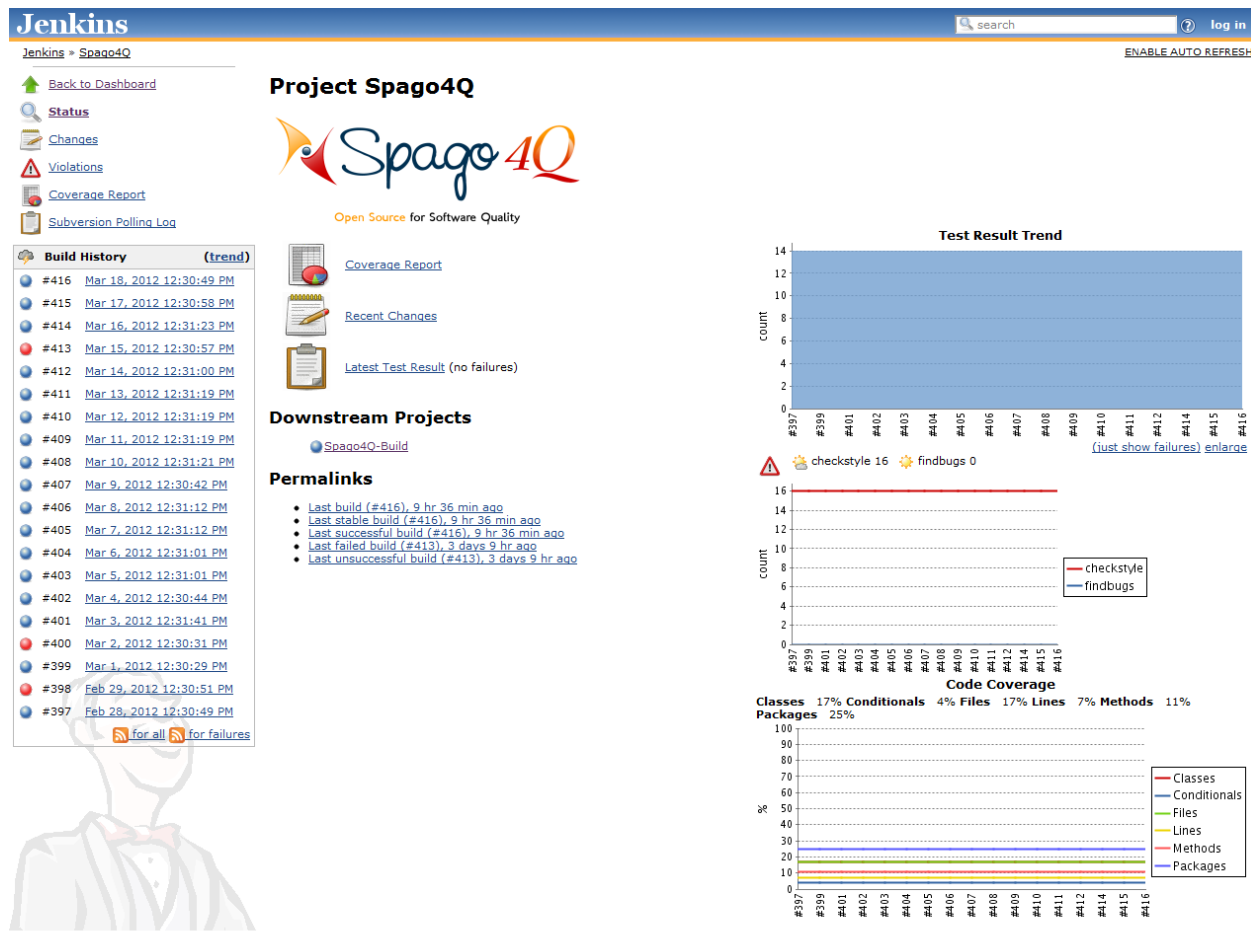
[Legend](#) [for all](#) [for failures](#) [for just latest builds](#)

- The main page provides a summary of the projects
- Quick view of
  - What's building ("No builds in the queue")
  - Build Executor Status (both "Idle")
  - Status of the projects


# Jenkins by example – Project Status

- Project status pages provide more details about a given project
  - The status of the last several builds
  - Charting (depending on plugins)
  - Dependencies

# Jenkins by example – Project Status



# Jenkins by example – New Project



Jenkins

test configuration

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

**Build History** (trend)

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

Project name: test

Description:

[Preview](#)

☐ Discard Old Builds

☐ This build is parameterized

☐ Disable Build (No new builds will be executed until the project is re-enabled.)

☐ Execute concurrent builds if necessary

**Advanced Project Options**

[Advanced...](#)

**Source Code Management**

☐ CVS

☒ None

☐ Subversion

**Build Triggers**

☐ Build after other projects are built

☐ Build periodically

☐ Poll SCM

**Build**

[Add build step](#)

**Post-build Actions**

☐ Aggregate downstream test results

☐ Archive the artifacts


☐ Build other projects

☐ Publish JUnit test result report

☐ Publish Javadoc

☐ Record fingerprints of files to track usage

☐ E-mail Notification



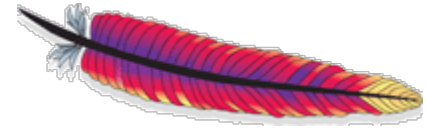
[Help us localize this page](#)

Page generated: Mar 10, 2012 4:00:44 PM Jenkins ver. 1.422

# Enhancing Jenkins

- Jenkins plugin system can enable a wide range of features including (but certainly not limited to)
  - SCM
    - Mercurial, Git, Subversion
  - Testing
    - Selenium, Windmill, TestLink
  - Notifications
    - IRC, Twitter, Jabber
  - Reporting
    - Doxygen, PMD, Findbugs
  - Artifact Saving
    - Artifactory, Amazon S3, SCP
  - Triggers
    - Jabber, Directory Watchers
  - External Integration
    - GitHub, Bugzilla, JIRA
  - And most importantly – The CI Game
    - A points based game where developers compete against each other to develop the most stable, well-tested code

# Who uses Jenkins?



**LinkedIn**

**ebay**

**CARFAX**  
VEHICLE HISTORY REPORTS

**NETFLIX**



**BOEING**

**mozilla**  
FOUNDATION



# Running Jenkins yourself

- Jenkins is packaged as a WAR, so you can drop it into whichever servlet container you prefer to use
- Jenkins comes pre-packaged with a servlet if you just want a light-weight implementation
- Native/Supported packages exist for
  - Windows
  - Ubuntu/Debian
  - Redhat/Fedora/CentOS
  - Mac OSX
  - openSUSE
  - FreeBSD
  - OpenBSD
  - Solaris/OpenIndiana
  - Gentoo

# Running Jenkins yourself – Updates

- Jenkins has two release lines
  - Standard releases
    - Weekly bug fixes and features
  - Long-Term Support releases
    - Updates about every 3 months
    - Uses a “Stable but older” version from the standard release line
    - Changes are limited to backported, well-tested modifications

# Letting someone else run Jenkins

- There are also cloud-based solutions that can provide a Jenkins instance
  - Cloudbees - <http://www.cloudbees.com/>
  - ShiningPanda - <https://www.shiningpanda.com/>

# Tying it into Agile

- For an Agile team, Jenkins provides everything needed for a robust continuous build system
- Jenkins supports Agile principles by constantly providing access to working copies of software
- Jenkins' extensibility allows the system to adapt to many different pre-existing environments

# Putting it all together

- While an integral part of a CI system, Jenkins is by no means the only component
- In order for a CI system to function, a common repository for the codebase needs to exist
- A database of artifacts needs to exist, so deliveries can be made at past iterations
- The last step in a CI process is the deployment of the components built
- ...and none of this matters if the developers don't use the system; procedures need to ensure the system is used as intended

# Conclusion

- Continuous integration is a necessity on complex projects due to the benefits it provides regarding early detection of problems
- A good continuous build system should be flexible enough to fit into pre-existing development environments and provide all the features a team expects from such a system
- Jenkins, a continuous build system, can be an integral part of any continuous integration system due to its core feature set and extensibility through a plugin system

# References

- Continuous Integration – Martin Fowler
  - <http://www.martinfowler.com/articles/continuousIntegration.html>
- Hudson
  - <http://hudson-ci.org/>
- Hudson Continuous Integration Server
  - <http://www.code-magazine.com/articleprint.aspx?quickid=0906071&printmode=true>
- The Hudson Book
  - <http://www.eclipse.org/hudson/the-hudson-book/book-hudson.pdf>
- Jenkins
  - <https://wiki.jenkins-ci.org>
- Monkey Image
  - [http://corrines-corner2006.blogspot.com/2011/09/freebie-monday\\_26.html](http://corrines-corner2006.blogspot.com/2011/09/freebie-monday_26.html)
- What is Continuous Integration
  - <http://confluence.public.thoughtworks.org/display/CCNET/What+is+Continuous+Integration>