SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

COURSE NAME : 19CS302 AGILE SOFTWARE ENGINEERING

II YEAR /III SEMESTER

Unit 2- Agile Development

Topic 9: Agile-Jenkins

Brain Storming

1. How to integrate different phases of a software product?

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

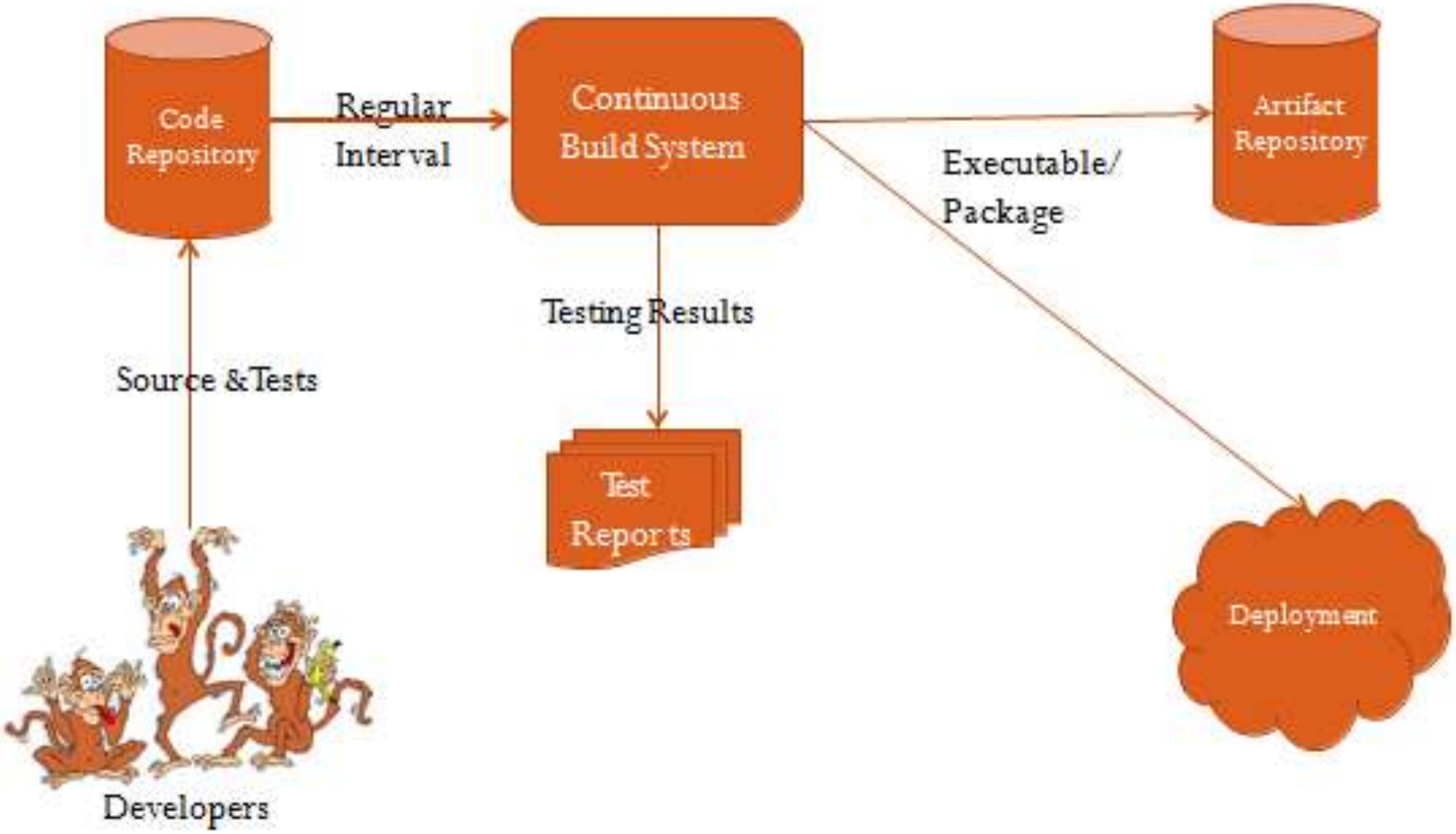
CI - Defined

- “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


Code Repository
Regular Interval
Continuous Build System
Artifact Repository
Executable/ Package
Testing Results
Source & Tests
Test Reports
Deployment
Developers

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

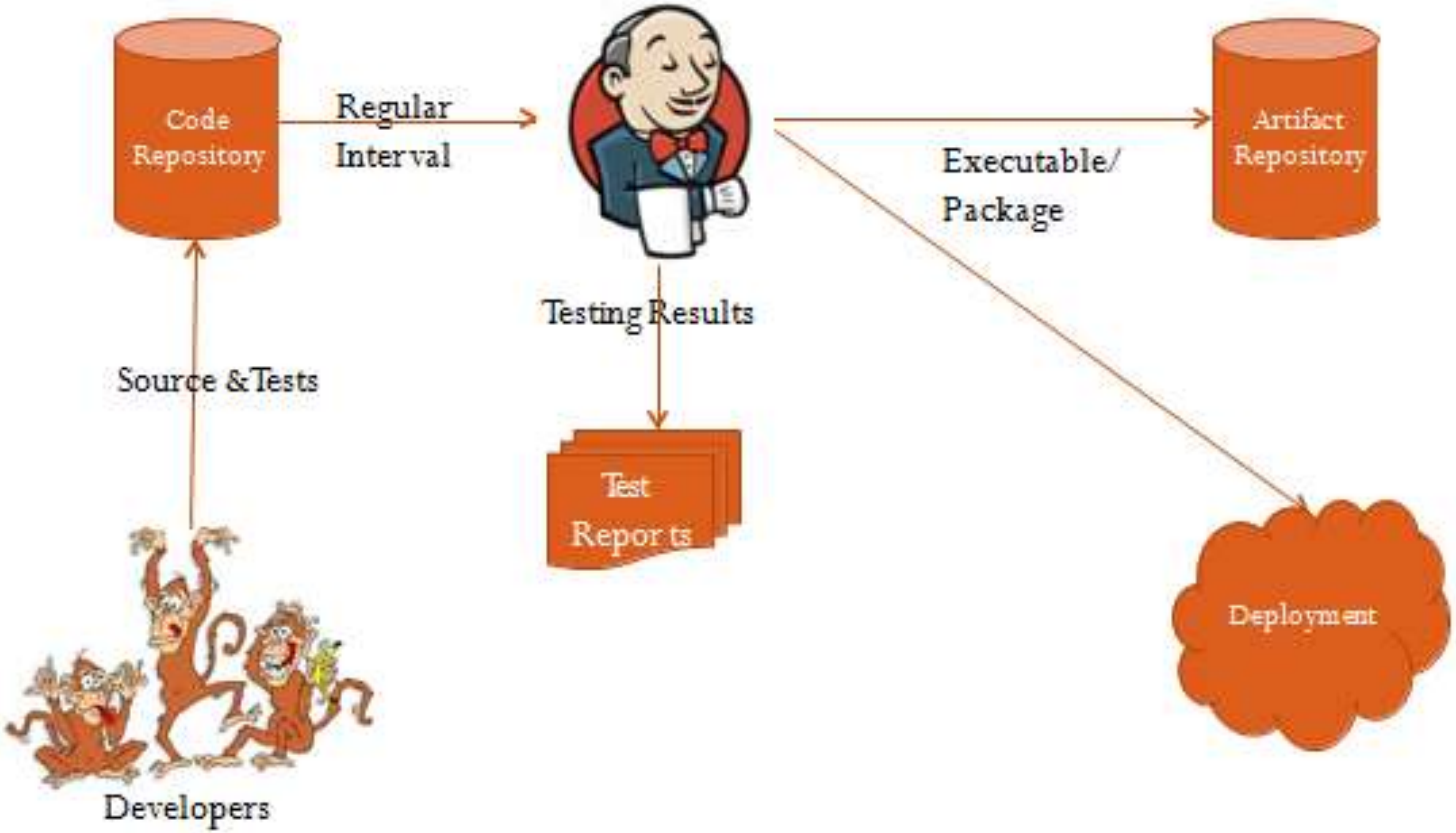
Jenkins

- 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

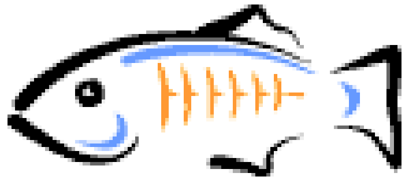
Why Jenkins? Award winning!

- 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

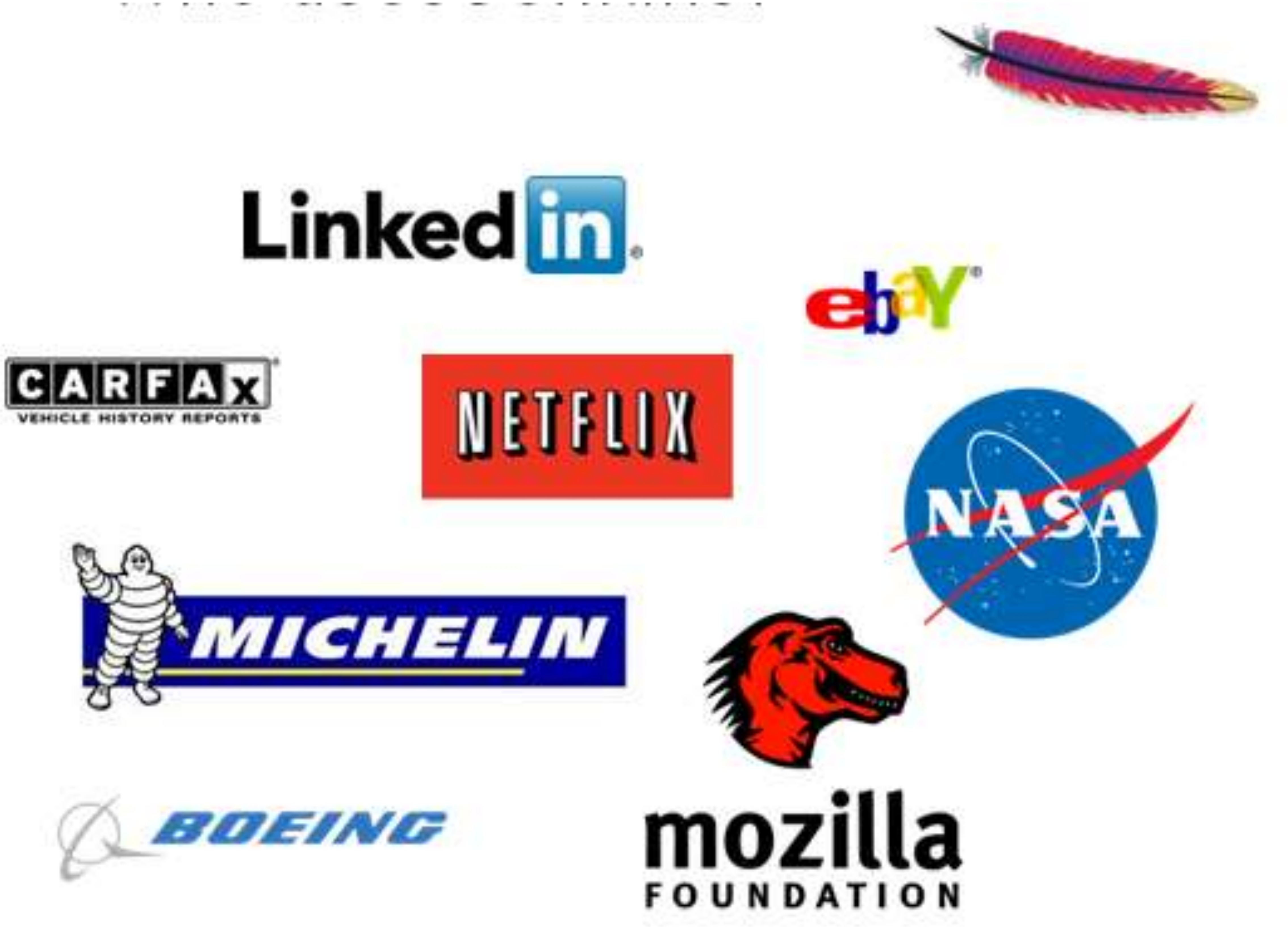
How Jenkins works - Building

- Once a project is successfully created in Jenkins, all future builds are automatic
- Building
  - Jenkins executes the build in an executer
    - By default, Jenkins gives one executer 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

Who uses Jenkins?



Summary

- 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 it’s core feature set and extensibility through a plugin system

Assessment 1

1. How Jenkins perform Integration process?

Ans : ____

2. List out the industries using Jenkins.

Ans : ____

References

1. Roger S.Pressman, Software engineering- A practitioner‘s Approach, 10th Edition, McGraw-Hill, 2017.
2. Ken Schawber, Mike “Agile Software Development with Scrum” Pearson Education, 2nd Edition, 2015.

Thank You