Continuous Integration & Code Quality

MINDS-ON

NUNO BETTENCOURT (NMB@ISEP.IPP.PT)

@DEI, 11 APRIL 2017
Continuous Integration

- THE THEORY -
Standard Software Development

• Individual code development

• Minimum code overlap
  ◦ Individuals focus on their “part” of the code

• Integration is performed manually

• Functional testing is achieved by means of manual procedures to ensure application is running
Continuous Software Development [1]

• Continuous Integration
  ◦ Continuously add new code commits
  ◦ Unit Testing & possibly Integration Testing

• Continuous testing
  ◦ Manual exploratory tests
  ◦ User acceptance testing

• Continuous delivery
  ◦ Functional Testing
  ◦ Regression Testing
  ◦ Pre-Generated Acceptance Tests
  ◦ Stage Environment

• Continuous deployment
  ◦ Production Environment
  ◦ Web & App Development

LAPR2, 3, 4
ODSOFT

ODSOFT

?
Continuous Integration

- THE TOOLS -
Basic Configuration
Continuous Integration

- THE VERSION CONTROL SYSTEM -
Git Repository [2]

• It is a distributed version/revision control system
• Focus on Data integrity
• Works in a Distributed Manner
• Allows non-linear workflows
  ◦ Centralized Workflow
  ◦ Feature Branch Workflow
  ◦ Gitflow Workflow
  ◦ Forking Workflow
Bitbucket [3] (i)

- **Team**
  - Big groups of users and repositories
    - Examples
      - lei-isep, mei-isep, dei-isep
  - Allows group definition of classes or students groups
    - Examples
      - LAPR2-2016-TEACHERS
      - LAPR4-2016-2DA
      - LAPR4-2016-TEACHERS

- **Project**
  - Dynamic groups of projects
  - One for each course edition, containing all students repositories
    - Examples
      - LAPR2-2016, LAPR3-2016

- **Repository**
  - Git Repositories
    - Example:
      - LAPR2-2016-G001, LAPR4-2016-2DA
Bitbucket (ii)

• Cloud Hosted
  ◦ https://bitbucket.org/ - Secure

• Project Template
  ◦ Group projects are forked from this one
  ◦ Forks must be performed MANUALLY for each group

• Users are added to repositories automatically
  ◦ Using REST API through a Jenkins Job using a Gradle script

• Future Work
  ◦ Automatically fork repositories based on a repository template
  ◦ Optimize Jenkins Job to automatically adds users to repositories
Bitbucket: Setup Workflow

- Create a Project for the Course
- Create a Template Git Repository and add it to the Project
  - Configure teachers’ permissions
- Fork the Template Repository for each group/class
  - Configure students’ permissions
Continuous Integration

- THE DEVELOPMENT PC -
Development PC

- Java Development Kit
  - JDK8

- Favourite IDE
  - NetBeans
  - Eclipse
  - IntelliJ Idea

- Normally included in any IDE
  - Maven Client
  - Git Client

- Browser
  - Jenkins
  - SonarQube
  - HipChat
  - Maven Repository

- HipChat Client
  - Optional

- Git Client
  - TortoiseGit, SourceTree, IDE Embedded
Apache Maven

• It is a software project management tool to manage a project's build, reporting and documentation
  ◦ Project Object Model (POM)

• Build Lifecycle
  ◦ validate - validate the project is correct
  ◦ compile - compile the source code
  ◦ test - test the compiled source code using unit testing framework
  ◦ package - package the compiled code in its distributable format e.g. JAR
  ◦ ..... 

• Maven Repository
  ◦ https://mvnrepository.com
Continuous Integration

- THE INTEGRATION SERVER -
Jenkins

• DEI Hosted
  ◦ https://jenkins.dei.isep.ipp.pt
  ◦ Secure, yet with self-signed certificate

• Allows the configuration of Jobs
  ◦ Each Job performs a set of pre-defined automatic tasks
    ◦ Maven tasks, Shell scripts, etc.

• Users are manually configured in Jenkins

• Users and permissions are manually configured for each Jenkins Job

• Future Work
  ◦ Automatically configure new users in Jenkins
  ◦ Automatically configure Jenkins Jobs’ users and permissions
Jenkins Plugins

- Most relevant plugins for this setup
  - Bitbucket Build Status Notifier Plugin
  - Bitbucket Plugin
  - Hipchat Plugin
  - SonarQube Scanner for Jenkins
  - PIT Mutation Plugin
Jenkins: Setup Workflow

- Create a Jenkins Template Job
  - Configure Bitbucket Repository
  - Configure Bitbucket Notify Build Status
  - Configure Teachers permissions
  - Configure SonarQube
  - Configure build process
    - Maven Build
    - Maven Test
    - Maven SonarQube Scanner

- Duplicate the Template Job for each group
  - Configure HipChat Notifications
  - Configure Students permissions

- Future Work
  - Automate HipChat Notifications configurations
Continuous Integration

- THE INTEGRATION PROCESS -
Bitbucket <-> Jenkins

• On Bitbucket
  ◦ For each Git Repository configure a Jenkins WebHook
    ◦ \texttt{https://jenkins.dei.isep.ipp.pt/bitbucket-hook/}
    ◦ Must skip certificate verification, because Jenkins@DEI uses a self-signed certificate
    ◦ Trigger: Repository Push
    ◦ MANUALLY configured for each repository
  ◦ Configure an SSH access for Jenkins to access the repository

• On Jenkins
  ◦ Add SSH credentials to access Bitbucket’s Git Repositories

• Future Work
  ◦ Automatic Webhook configuration tool
  ◦ Add a valid certificate to Jenkins
Workflow
Code Quality
- THE THEORY -
Code Quality

• Software Functional Quality
  ◦ reflects how well the code complies with or conforms to a given design
  ◦ based on functional requirements or specifications

• Software structural quality
  ◦ refers to how it meets non-functional requirements that support the delivery of the functional requirements
  ◦ e.g. robustness, maintainability, complexity, etc.
  ◦ the degree to which the software was produced correctly
Code Quality

- THE TOOLS -
Code Quality: Basic Configuration
Code Analysis

• Unit Testing & PIT Mutation Testing
  ◦ Coverage

• Static Code Analysis
  ◦ Code Analysers
  ◦ PMD
  ◦ CheckStyle
  ◦ FindBugs
  ◦ Code Duplication
  ◦ Complexity
  ◦ Documentation
SonarQube

• DEI Hosted
  ◦ https://sonarqube.dei.isep.ipp.pt
  ◦ Secure, yet with self-signed certificate

• Dashboards
  ◦ Technical Debt
  ◦ Code Coverage
  ◦ Code Duplication
  ◦ Issues
  ◦ Quality Gate
  ◦ Code Structure
  ◦ PIT Mutation Testing
SonarQube Plugins

- C#
- CheckStyle
- Cobertura
- FindBugs
- GIT
- Jira
- JAVA
- JavaScript
- PMD
- SVN
- PIT
SonarQube: Setup Workflow

• Create one Project for each Jenkins Job
  ◦ Preferably with the same name

• For each repository configure users permissions
  ◦ Students should be configured as a group
  ◦ Teachers should be configured as a group
  ◦ Configure the students and teachers’ group access to the project by adding them “Browse”, “See Source Code”, “Administer Issues” and “Execute Analysis” permissions
  ◦ Configure exclusion rules (e.g. Code Coverage on UI)

• Future Work
  ◦ Solve a bug in the workflow that requires to manually add each group
  ◦ Improve the current Excel/SQL scripts generator tool
Workflow

1. Commit
2. Trigger Build
3. Request Repository
4. Request Code Quality Profile
5. Publish Code Quality Report

LAPR2, 3
Issue Management - THE TOOLS -
Jira

• DEI Hosted
  ◦ Insecure: Lacks HTTPs, passwords sent in clear text over insecure network (e.g. ISEPWLAN)

• Issue Management
  ◦ When used in combination with Agile (Scrum), each issue describes a User Story
  ◦ Each Issue is given a unique identifying token

• Allows the tracking of code changes to solve an issue
  ◦ By using the token in the commit message
Workflow
Notifications & Teamwork

- THE TOOLS -
Hipchat

- Cloud Hosted
  - [https://www.hipchat.com](https://www.hipchat.com)
  - Secure
- Each group is assigned a room
- Rooms are manually created
- Users are manually added to rooms
- Jenkins build Notifications
  - Success
  - Failed
- Future Work
  - Automatic creation of rooms
  - Automatically adding users to rooms
Bitbucket Notifications

- Build Notifications for Repositories
Workflow
There’s more...

• GitInspector
• Dependency Structure Matrix (DSM)
• Code Review
• ...
Questions?
References

• 1 - http://searchsoftwarequality.techtarget.com/definition/Continuous-Software-Development

• 2 - https://en.wikipedia.org/wiki/Git