

ECSE 321
Introduction to Software Engineering

Deliverable 4
Release Pipeline Plan

March 29, 2018

Group 11

Alexander Harris - 260 688 155
Abbas Yadollahi - 260 680 343
Filip Bernevec - 260 689 062
Yunus Can Cukran - 260 669 715
Gareth Peters - 260 678 626

Table of Contents

Table of Contents	2
INTEGRATION PHASE	3
Description	3
Tools Used	3
Git/GitHub	3
Jenkins	4
JUnit	4
EclEmma	4
Discord	4
Tools Considered	5
Slack	5
BitBucket	5
Design	5
Rationale	5
BUILD PHASE	5
Description	5
Tools Used	5
Design	5
Rationale	5
DEPLOYMENT PHASE	6
Description	6
Tools Used	6
Design	6
Rationale	6
WORK PLAN	6

INTEGRATION PHASE

Description

The integration pipeline allows for continuous integration of new modules and breaks down these modules into stages. Each stage is aimed at verifying the quality of new features from a different angle to validate the new functionality and prevent errors. Each new module is continuously integrated and unit tested to reduce the risk of any errors ever reaching the user. The integration pipeline also allows for version management, so any change can be reverted if there is ever a mistake.

Tools Used

This section describes the tools that were used in the integration pipeline. The tools used include: Git, GitHub, Jenkins, JUnit, EclEmma, Discord, and Google Drive.

Git/GitHub

Git is a distributed version control system that allows our team to keep track of our code. GitHub is the website that hosts the source code and allows for our team to easily review, manage, and build our software.

Git allows our team to easily access and make changes to our software. It also allows us to track who has made the changes so one can revert back to a previous version of the code to fix any potential bugs or mistakes. With the use of branching, many people can work on different modules in the system at a time without merging conflicts.

GitHub then acts as source code management for our Git repository. It acts as a failsafe in case of accidental deletions of code, or even worse, the repository itself. IT also allows us to review statistics of the repository. An example of this is the use of graphs to check the work distribution of the team to see who is not putting in enough work and who is putting in too much work.

Git is an industry-standard version control tool used by many in the workfield and no alternatives were considered. We did not have much choice in using GitHub instead of other hosting services. Our repository on GitHub is reviewed by those in charge of ECSE 321. However, even if it were our choice we still would have used GitHub because it is the most reliable and has excellent integration with Jenkins.

Jenkins

Jenkins is a continuous integration service that reacts to each new version of source code pushed onto GitHub. This reaction is to build the project and run tests on the new version of the source code. This allows for test automation and ensures that the code functions before reaching the user.

Jenkins was used instead of other continuous integration services, like Travis-CI, because we are provided a dedicated server by the ECSE 321 staff. Jenkins usually requires setting up a dedicated server, whereas Travis-CI offers free hosting services. Setting up a dedicated would usually be an extra expense otherwise. Jenkins also allows hosting for a private project for free, where Travis-CI involves a monthly payment to host private projects .

JUnit

EclEmma

EclEmma is used in conjunction with JUnit. It is a plugin for Eclipse that provides a testing coverage reports. Our team decided to utilize this program so as to maximize the amount of coverage in our system.

Our team chose EclEmma because of the readability of the coverage reports. It does not take very long for someone to understand how EclEmma works and is easily accessible to anybody using Eclipse. The coverage reports can also be exported as PDFs for versioning. Lastly, EclEmma is able to provide coverage for separate instructions in the software, whereas some other coverage tools like Cobertura and Atlassian Clover do not.

Discord

Discord is a powerful communication tool that allows our team to keep in contact. Although it is meant mostly for gaming, the use of Discord allows us to have meetings through voice-chat, and discuss deliverables and the application on the go. Discord servers allow for multiple voice and text chats, which allowed us to easily organize questions and queries any team member might have with respect to, but not limited to, the backend services, Android frontend or Web frontend.

Discord was chosen over other communication services like Facebook messenger and Slack mainly because of the voice-chat capability. Our team is always able to quickly answer one another's questions without any loss in translation created when asking questions through writing.

Tools Considered

Slack

BitBucket

Design

Rationale

BUILD PHASE

Description

Tools Used:

Web:

Webpack

Npm

Backend:

gradle

Design

Rationale

DEPLOYMENT PHASE

Description

Tools Used

Design

Rationale

WORK PLAN