# Project Report 1

Frameworks to Support Continuous Delivery

with Kostas Kangaro

Gurpreet Singh & Paul Bartlett

*# 250674134 & 250123456*

*Software Developers*

# Contents

# CS4470Y

## Project description

shortening bug fixing shortening maintenance cycles Achieving continuous delivery

## Roles

**Gurpreet Singh**

- Lead Architect
- Documenter

**Paul Bartlett**

- Project Manager
- Lead Requirements Analyst
- Lead Tester & Quality Controller

## System Requirements

**Section A : Data collection and modeling**

- **Feature 1:** Git Based Data Collection
- FR 1: Extract commits from a .git directory
- FR 2: Analyze and record classes/methods effected by each commit
- FR 3: Process and record the amount of code changed with each commit
- **Feature 2:** Test Based Data Collection
- FR 1: Record results of the test suite each time it is run
- FR 2: During test phase inject previous risk analysis
- **Feature 3:** Issue Based Data Collection

- FR 1: Record issues filed for the code base
- FR 2: Connect issues to their resolution and make an inference from the association
- **Feature 4:** Data combination
- FR 1: Standardize and combine Git data into a central data model
- FR 2: Standardize and combine test data into a central data model
- FR 3: Standardize and combine issue data into a central data model

**Section B : Creating report processing software**

- **Feature 4:** Conduct Risk Analytics
- FR 1: In the data model figure out risky parts of the code and mark them
- FR 2: Identify issue keywords that are higher risk than others
- FR 3: Identify impact when a test is changed (how much coverage it offers)
- **Feature 5:** User Views
- FR 1: Create a PDF report for stakeholders to see progress in codebase
- FR 2: Create a Web view for exploring the risk of the codebase
- FR 3: Create Github, IFTTT and Slack integrations for viewing framework data model

**Section C : Creating enhancements to dev environments**

- **Feature 6:** Continuous Integration
- FR 1: Create test integration so developer can see risk rating when conducting tests
- FR 2: Develop command line interface to the framework
- FR 3: Research deployment frameworks and integrate risk analysis into pipe line
- **Feature 7:** IDE integration

- FR 1: Develop VIM plugin for viewing risk rating of each method/class
- FR 2: Develop IDEA plugin for viewing risk rating of each method/class
- **Feature 8:** Microservice Plug-And-Play library
- FR 1: Make the framework easily deployable into a microservice
- FR 2: Reduce dependencies and deliver a minimally intrusive product