COMPX341-23A Assignment Four: Software Maintenance and DevOps

Due date: 11:59pm Friday 2nd June, 2023
Submission: LaTeX PDF via Moodle and Software via UoW GitLab
Weight in overall grade: 20%

Abstract

You work as a Software Engineer for SoftFlux, a software engineering company. A new and emerging Smart Home development company, called Encost, has approached your company with a proposed project called the Encost Smart Graph Project (ESGP). ESGP is a software system that enables the visualisation of Encost's devices using a graph data structure.

Introduction

You still work as a Software Engineer for SoftFlux, a software engineering company. While you were busy implementing your software, a different team was working on an alternative implementation, which the client has accepted. The system has now been deployed and released to the public. While the original SRS stated that only the initial development of the software was covered, your company (SoftFlux) and the client (Encost) have agreed that SoftFlux will continue on with a Maintenance contract. Your task now is to continue on with the project as a Software Maintainer and DevOps Engineer.

As part of your new role, your team lead has given you the following three maintenance-based user stories.

1. *"The client (Encost) has decided that the system should use the most up-to-date version of GraphStream"*
2. *"The Encost users have noticed that the summary statistics aren't working. They would like to be able to see a textual summary of the device distribution (SRS 4.9 Calculating Device Distribution)"*
3. *"The Community users are having trouble telling which household is theirs in the graph visualisation. They would like the additional feature of being able to see a graph visualisation for their household only"*

Five versions of the software (along with their corresponding SRS, SDS, and Test Plan documents) will be uploaded to Moodle. You must select one of these implementations as the 'client accepted version'. Please note, you cannot select your own document. Overall, in the assignment, you are required to do the following.

1 Complete a Maintenance document

A LaTeX template has been provided. Your task is to create a Maintenance document outlining the details of the three maintenance tasks listed above, including an Impact Analysis and your proposed solution for each. For each maintenance task, your Maintenance document should include the following.

- The maintenance user story as stated in this specification
- The problem/modification request
- The problem/modification analysis
  - Determine the maintenance category (preventative, perfective, corrective, adaptive)
  - Conduct an informal Impact Analysis evaluation
- The modification request acceptance/rejection
  - Determine whether to accept or reject the request
  - Please note, you may reject **one** of the maintenance tasks
- Modification implementation
  - If the maintenance task was accepted, then you are required to implement it
  - You should include evidence of your implemented solution in the document, whether this be a textual description, a diagram, a screenshot, or a combination of all three.
- Show evidence of maintenance review / acceptance
  - E.g. testing and/or code review
  - If the maintenance task was accepted, then you are required to test it
  - You should include evidence of your testing in the document, whether this be a description of manual tests (with screenshots) or a description of a unit test(s). If unit tests are developed they should be uploaded to GitLab.

2 Develop a CICD Pipeline

Up until now you have implemented all of your commands (for compilation, testing, delivery, and execution) manually. Part Two of the assignment requires you to write a CI/CD pipeline. Your pipeline should run under the same Operating Environment as outlined in Section 2.3 of the Software Requirement Specifications document. Your pipeline should be submitted to GitLab within the top level of your Software repository, and should include the following.

1. Prepare (some form of preparation, if needed)
2. Build (compiling the software)
3. Test (e.g. running regression tests)

   (only if compilation was successful)
4. Release (committing to a repository)

   (only if all tests passed)
5. Deploy (in our case, just the java run command)

   (only if release was successful)

Notes

1. You are required to submit your software (including the CICD Pipeline) using UoW GitLab (`https://courses-git.cms.waikato.ac.n`). Please ensure that you commit to GitLab (providing *good* commit messages) throughout, not just at the end.
2. The use of the Maintenance document is uncommon in industry. The template is used in this assignment to give you some experience working with Impact Analysis.