ACIT 4850 – Enterprise Systems Integration – Assignment 2

Instructor	Mike Mulder (mmulder10@bcit.ca or Slack)
Total Marks	20
Due Dates	Written - March 20th, 2020 at midnight on D2L Demo – Week of March 23rd in class or during office hours (no later than 1pm on Thursday, March 26th

This assignment is to be completed and submitted individually.

Overview

This time you already have a solution for the tool you are evaluating in your Enterprise Development Environment. But your software development team is looking at alternative new options – because developers like trying out new tools – in the following areas:

- Source Code Management
- Continuous Integration
- Static Code Analysis

Options

For this assignment, choose one of the following three options for your assessment.

Option 1 – Source Code Management

We already have GitLab in our Enterprise Development Environment (as of Lab 1) for source code management. Evaluate and prototype one competing on-premise options against GitLab, such as:

- GitHub
- Bitbucket
- Gitolite
- Etc.

You must be able to reproduce the integrations we have with Jenkins builds with your given choice. This means you can demonstrate that a Python and Java pipeline job can be integrated – the jobs retreive the code from the repo and are triggered by code changes.

Note: It is your responsibility to make sure the one you choose can be sufficiently compared against GitLab.

Option 2 – Continuous Integration

We already have Jenkins in our Enterprise Development Environment (as of Lab 4) for Continuous Integration. Evaluate and prototype one competiting on-premise or cloud options against Jenkins, such as:

- GitLab CI
- Circle CI
- Travis CI
- Azure DevOps
- Etc.

You must be able to reproduce the integrations we have with GitLab projects (i.e., pull from repository and trigger builds on pushes to the repository) and the existing build pipelines we have on Jenkins (one Python and one Java).

Note: It is your responsibility to make sure the one you choose can be sufficiently compared against GitLab.

Option 3 – Source Code Quality

We already have SonarQube in our Enterprise Development Environment (as of Lab 7) for Static Code Analysis. Evaluate and prototype one competiting on-premise or cloud options against SonarQube, such as:

- Checkmarx
- GitLab (built-in code analysis)
- Etc.

You must be able to reproduce the integrations we have with the Jenkins CI Pipelines for Python and Java applications (i.e., integrate code analysis stages).

Note: It is your responsibility to make sure the one you choose can be sufficiently compared against SonarQube.

Assessment

Demonstration (12 marks)

- Demo your selected tool integrated into your Enterprise Development Environment
    - Souce Code Management – Integrated with Jenkins and same 2 repos as GitLab (one Python, one Java)

- Continuous Integration – Integrated with GitLab and same pipeline builds as Jenkins (one Python, one Java)
- Source Code Quality – Integrated code analysis with the Build Pipelines in Jenkins (one Python, one Java)

Written Assessment (8 marks)

This assessment should be based on your prototype and hands-on experience with the tool. It should NOT be the same as your Assignment 1 comparison.

Your written submission should include the following content:

- Comparison – Qualitatively compare and constrast our existing tool with your selected tool based on your prototype. This part should be written (i.e., a paragraph).
- Detailed Comparison – Use a table again (similar to Assignment 1) to do a qualitative comparison of the two tools based on your prototype.
- Recommendation – Identify which you would choose and why. Also describe why you would not choose the other tool OR in what circumstances you would use the other tool.
- References – For any online research used in your written submission.

The written portion must be 1.5-3 pages in length, excluding any title pages and references. The submission should be in **PDF format** and *does not need to originate from Confluence*.

Submission

Demo – Demonstrate your submission in class or office hours during the week of March 23th. No later than March 26th at 1pm.
Written – Submit your PDF to the D2L dropbox (Activities -> Assignments -> Assignment 2) by March 20th at midnight.

Grading Summary

Demonstration • Equivalent configuration to the existing tool (9 marks) • Answering 3 questions related to the implementation (3 marks)	12 marks
Written • Qualitative Comparison (2 marks) • Quantitative Comparison (2 marks) • Recommentation (4 marks)	8 marks

Make sure you reference any sources (i.e., websites) you used for your assessment. Marks will be subtracted for no references.	(-2 marks)
Make sure your submission is in PDF format	(-2 marks)
Total	**20 marks**

Written Assessment

Levels of Mastery	**Level 1**	**Level 2**	**Level 3**	**Level 4**
	A limited mastery of knowledge and skills: below basic expectations	A partial mastery with limited to basic performance of expected achievement	A solid consistent performance; demonstrated competency of knowledge and skills	A superior, consistent performance; beyond expectations
	Major Problems Exist	**Minor Problems Exist**	**Minor Issues Exist**	**No Issues Exist**
Description, Requirements and Summary of Each Tool	Not evident	Description, Requirements and Summary of Each Tool are not thorough or complete and required sections are not included. No unique elements.	Description, Requirements and Summary of Each Tool are mostly thorough or complete and all required sections are included. Mostly unique elements.	Description, Requirements and Summary of Each Tool are thorough or complete and all required sections are included. The content is highly refined. All elements are unique.
Detailed Comparison Table	Not evident	Limited comparison is done across the required categories. Some categories are missing or have minimal content. No sources are cited. Minimal research was done.	Comparison is mostly thorough and shows research was done into each of the tools. Most sources are cited. Research is presented adequately.	Comparison is thorough and shows detailed research into each of the tools. All sources are cited. Research is presented appropriately.
Recommendation and Justification	Not evident	No recommendation and/or justification is provided for a tool.	A recommendation is provided with justification. It mostly draws upon the research and analysis from the previous sections. Some justification for not selecting the other tools is given.	A recommendation is provided that draws upon the research done for the previous sections. Further research on other possible candidate tools is provided. Detailed justification for not selecting the other tools is provided that again draws upon the research done.
	0-20%	**20-50%**	**50-90%**	**90-100%**