Craig Rowell

2022/12/10

CS-320 Software Test Automation & QA

Project Two: Summary and Reflection

# Summary

## Testing Approach

* 1. Describe your unit testing approach for each of the three features.

### Alignment to Requirements

* + 1. To what extent was your approach **aligned to the software requirements**

### Effective Tests

* + 1. Defend the overall quality of your JUnit tests. In other words, how do you know your JUnit tests were **effective** based on the coverage percentage?

## Testing Experience

* 1. Describe your experience writing the JUnit tests.

### Technically Sound Code

* + 1. How did you ensure that your code was **technically sound**? Cite specific lines of code from your tests to illustrate.

### Efficient Code

* + 1. How did you ensure that your code was **efficient**? Cite specific lines of code from your tests to illustrate.

# Reflection

## Testing Techniques

### Techniques Employed

* + 1. What were the **software testing techniques** that you employed in this project? Describe their characteristics using specific details.

### Other Techniques

* + 1. What are the **other software testing techniques** that you did not use for this project? Describe their characteristics using specific details.

### Uses and Implications of Techniques

* + 1. For each of the techniques you discussed, explain the **practical uses and implications** for different software development projects and situations.

## Mindset

### Caution

* + 1. Assess the mindset that you adopted working on this project. In acting as a software tester, to what extent did you employ **caution**? Why was it important to appreciate the complexity and interrelationships of the code you were testing? Provide specific examples to illustrate your claims.

### Bias

* + 1. Assess the ways you tried to limit **bias** in your review of the code. On the software developer side, can you imagine that bias would be a concern if you were responsible for testing your own code? Provide specific examples to illustrate your claims.

### Discipline

* + 1. Finally, evaluate the importance of being **disciplined** in your commitment to quality as a software engineering professional. Why is it important not to cut corners when it comes to writing or testing code? How do you plan to avoid technical debt as a practitioner in the field? Provide specific examples to illustrate your claims.