## Competencies

In this project, you will demonstrate your mastery of the following competencies:

- Analyze various approaches to software testing based on requirements
- Apply appropriate testing strategies to meet requirements

## Scenario

You are a software engineer for Grand Strand Systems, a software engineering company that focuses on developing and testing back-end services. You recently completed an assignment in which you developed a mobile application for a customer and delivered the contact, task, and appointment services (Project One).

You will now construct a summary and reflections report to be submitted to your supervisor that summarizes your unit testing approach, your experience writing the JUnit tests, and the overall quality of your JUnit tests. This report will also highlight testing techniques and the mindset you adopted while working on this project.

## Directions

### Summary and Reflections Report

Your supervisor has asked that you submit a follow-up summary and reflections report to explain how you analyzed various approaches to software testing based on requirements and applied appropriate testing strategies to meet requirements while developing the mobile application for the customer. This report should be based on your experience completing Project One. You must complete the following:

1. **Summary**
    a. Describe your unit testing approach for each of the three features.
        i. To what extent was your approach **aligned to the software requirements**? Support your claims with specific evidence.
        ii. 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?
    b. Describe your experience writing the JUnit tests.
        i. How did you ensure that your code was **technically sound**? Cite specific lines of code from your tests to illustrate.
        ii. How did you ensure that your code was **efficient**? Cite specific lines of code from your tests to illustrate.
2. **Reflection**
    a. Testing Techniques
        i. What were the **software testing techniques** that you employed in this project? Describe their characteristics using specific details.
        ii. What are the **other software testing techniques** that you did not use for this project? Describe their characteristics using specific details.
        iii. For each of the techniques you discussed, explain the **practical uses and implications** for different software development projects and situations.
    b. Mindset
        i. 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.
        ii. 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.
        iii. 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.

## What to Submit

To complete this project, you must submit the following:

### Summary and Reflections Report

Your summary and reflections report should be a 3- to 4-page Word document with 12-point Times New Roman font, double spacing, and one-inch margins. Sources should be cited according to APA style.

**Project Two Rubric**

| Criteria | Exemplary (100%) | Proficient (85%) | Needs Improvement (55%) | Not Evident (0%) | Value |
|---|---|---|---|---|---|
| **Summary: Alignment to Requirements** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Defends the alignment of the testing approach to the software requirements with specific examples from the test plan | Shows progress toward proficiency, but with errors or omissions; areas for improvement may include additional details or clarification defending the alignment of the testing approach | Does not attempt criterion | 8.5 |
| **Summary: Effective Tests** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Defends the effectiveness of the tests by reflecting on the coverage | Shows progress toward proficiency, but with errors or omissions; areas for improvement may include additional details or clarification defending the effectiveness of the tests | Does not attempt criterion | 8.5 |
| **Summary: Technically Sound Code** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Describes test writing experience, including strategies used to ensure technically sound code, and illustrates each with specific examples from the code | Shows progress toward proficiency, but with errors or omissions; areas for improvement may include additional details, clarification, or illustrations to ensure the code was technically sound | Does not attempt criterion | 8.5 |
| **Summary: Efficient Code** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Describes test writing experience, including strategies used to make the code efficient, and | Shows progress toward proficiency, but with errors or omissions; areas for improvement may | Does not attempt criterion | 8.5 |

| | | | | | |
|---|---|---|---|---|---|
| | | illustrates each with specific examples from the code | include additional details, clarification, or illustrations to ensure the code was efficient | | |
| **Reflection: Techniques Employed** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Describes the characteristics of the testing techniques employed in the project and uses specific details | Shows progress toward proficiency, but with errors or omissions; areas for improvement may include additional details or clarification describing the characteristics of the testing techniques | Does not attempt criterion | 8.5 |
| **Reflection: Other Techniques** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Describes the characteristics of the other testing techniques and uses specific details | Shows progress toward proficiency, but with errors or omissions; areas for improvement may include additional details or clarification describing the characteristics of other testing techniques | Does not attempt criterion | 8.5 |
| **Reflection: Uses and Implications of Techniques** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Explains the practical uses and implications of each of the techniques and uses specific examples of different software development projects and situations | Shows progress toward proficiency, but with errors or omissions; areas for improvement may include additional details or clarification explaining the practical uses and implications of each of the techniques | Does not attempt criterion | 8.5 |
| **Reflection: Caution** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Assesses the use and importance of employing caution when testing code and uses specific supporting examples | Shows progress toward proficiency, but with errors or omissions; areas for improvement may include additional details or clarification assessing the use and importance of employing caution | Does not attempt criterion | 8.5 |

| | | | | | |
|---|---|---|---|---|---|
| | | | when testing code | | |
| **Reflection: Bias** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Assesses the use and importance of limiting bias when testing code and uses specific supporting examples | Shows progress toward proficiency, but with errors or omissions; areas for improvement may include additional details or clarification assessing the use and importance of limiting bias when testing code | Does not attempt criterion | 8.5 |
| **Reflection: Discipline** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Evaluates the importance of being disciplined when developing code and uses specific supporting examples | Shows progress toward proficiency, but with errors or omissions; areas for improvement may include additional details or clarification evaluating the importance of being disciplined when developing code | Does not attempt criterion | 8.5 |
| **Articulation of Response** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Clearly conveys meaning with correct grammar, sentence structure, and spelling, demonstrating an understanding of audience and purpose | Shows progress toward proficiency, but with errors in grammar, sentence structure, and spelling, negatively impacting readability | Submission has critical errors in grammar, sentence structure, and spelling, preventing understanding of ideas | 10 |
| **Citations and Attributions** | Uses citations for ideas requiring attribution, with few or no minor errors | Uses citations for ideas requiring attribution, with consistent minor errors | Uses citations for ideas requiring attribution, with major errors | Does not use citations for ideas requiring attribution | 5 |
| | | | | **Total:** | 100% |