ASSESSMENT 3 - BRIEF	
Subject Code and Title	SDM404 Software Development and Management
Assessment	Project Test Plan
Individual/Group	Group/Collaborative
Length	Project Test Plan, Maximum 5 pages (+/- 10%) + Individual Contribution Report, 500 words (+/- 10%)
Learning Outcomes	This assessment addresses the Subject Learning Outcomes outlined at the bottom of this document.
Submission	Due by 11:55pm AEST Sunday of Module 10.
Weighting	15%
Total Marks	100

Context

Software Test Plan is a crucial document that outlines the strategy, scope, objectives, and approach for testing a software application. Its primary purpose is to ensure that the software meets the required standards and functions correctly before it is released. It helps teams identify potential issues early, define testing procedures, allocate resources efficiently, and establish clear timelines. By following a well-structured test plan, developers and testers can systematically verify that the software is free of defects, performs as expected, and delivers a smooth user experience.

For students learning about software development, understanding the importance of a test plan is essential, as it fosters good testing practices and improves the overall quality and reliability of software projects.

Task Instructions

To complete this assessment, you need to write and submit a Project Test Plan (PTP) along with the Defect Register. The test plan should include the following information about the testing of your project.

- Introduction
- Test Objectives

- Test Scope
- Test Strategy
- Test Environment
- Test Items
- Test Cases
- Test Results
- References
- Glossary

Please see the Project Test Plan Template provided by your Learning Facilitator, for the details of each section.

Submission Instructions

Submit your STP document in pdf format in Turnitin via the Assessment link in the main navigation menu in SDM404 Software Development Management. The Learning Facilitator will provide feedback via the Grade Centre in the LMS portal. Feedback can be viewed in My Grades.

Referencing

If needed, it is essential that you use appropriate APA style for citing and referencing research. Please see more information on referencing here http://library.laureate.net.au/research_skills/referencing

Assessment Rubric

Assessment Attributes	High Distinction (Exceptional) 85-100%	Distinction (Advanced) 75-84%	Credit (Proficient) 65-74%	Pass (Functional) 50-64%	Fail (Yet to achieve minimum standard) 0-49%
Test Objectives **10 %**	Exceptionally clear, precise, and detailed test objectives. Fully aligned with all relevant software requirements and user needs. Objectives are measurable, actionable, and contribute to a structured testing process. Demonstrates critical thinking in defining comprehensive and meaningful objectives.	Well-defined and specific test objectives. Strong alignment with software requirements and user expectations. Clearly measurable goals with expected results.	Clear and relevant test objectives. Objectives align with key software requirements. Some measurable outcomes are included, though not fully comprehensive.	Basic test objectives are stated but lack clarity. Some connection to software requirements, but limited detail. Objectives may not be fully measurable or specific.	Test objectives are unclear, missing, or irrelevant. No connection to software requirements or user needs. No measurable goals or expected outcomes.
Test Scope ***10%***	Exceptionally clear and comprehensive test scope. Precisely defines all features, modules, and	Well-structured and detailed test scope. Clearly differentiates between what is included and excluded from testing.	Clearly defines what features, functionalities, and components will be tested. Some aspects of exclusions are mentioned but may lack detail.	Basic test scope is provided but lacks clarity or depth. Some mention of included and excluded features, but lacks detail.	Test scope is missing, unclear, or irrelevant. No distinction between what is included and excluded from testing.

	functionalities that will be tested. Clearly justifies what is excluded from testing and why. Demonstrates a deep understanding of project requirements and testing priorities.	Strong alignment with project requirements and objectives.	Shows a reasonable connection to project requirements.	Limited connection to project requirements.	Lacks connection to project requirements.
Test Strategy ***10%***	Exceptionally well-structured and comprehensive test strategy. Clearly outlines testing methodologies, levels, and techniques with strong justification. Details automation vs. manual testing approaches with reasoning. Addresses risk-based testing, test prioritization, and resource allocation effectively. Demonstrates a deep understanding of best practices in software testing.	Well-defined test strategy with clear methodology. Includes detailed testing levels (unit, integration, system, acceptance, etc.). Specifies appropriate testing techniques (black-box, white-box, regression, etc.). Discusses automation strategy, if applicable.	Clearly defines a structured approach to testing. Mentions different testing methods and their relevance. Some explanation of testing levels, types, and tools used.	Basic test strategy is provided but lacks depth. Some mention of testing methods (e.g., manual vs. automated) but not well explained. Limited or unclear testing levels (unit, integration, system, etc.)	Test strategy is missing, unclear, or lacks relevance. No defined approach for testing. No mention of testing techniques, levels, or methodologies.

Test Environment ***10%***	Exceptionally detailed and comprehensive test environment description. Covers all technical aspects, including hardware, software, network, test data, virtualization, and cloud-based testing (if applicable). Specifies configuration settings, system dependencies, and any required third-party integrations. Addresses environment constraints, version control, and rollback mechanisms. Demonstrates a strong understanding of the importance of a well-prepared testing environment.	Well-defined and structured test environment details. Covers all key aspects, including hardware specifications, operating systems, test data, network configurations, and dependencies. Discusses environment setup procedures and tools used.	Clearly defines test environment setup, including hardware and software requirements. Mentions network, database, or third-party dependencies where applicable. Some explanation of configuration and tools used for testing.	Basic test environment is described but lacks detail. Some mention of hardware, software, or network configurations, but not comprehensive. Limited explanation of dependencies or setup requirements.	Test environment details are missing, unclear, or irrelevant. No mention of hardware, software, or network requirements. Lacks connection to testing needs.
Test Items ***10%***	Exceptionally clear, detailed, and well-organized list of test items.	Well-structured and detailed list of test items. Includes all key software modules, features, and	Clearly defines test items, including major features, modules, or components to be tested.	Basic test items are listed but lack detail. Some mention of features or modules to be tested, but	Test items are missing, unclear, or irrelevant. No clear identification of what components or

	Covers all critical software components, modules, and features in detail. Clearly maps test items to software requirements, use cases, or specifications. Considers various testing perspectives, including functional, non-functional, regression, and edge cases. Demonstrates a strong understanding of test coverage and prioritization.	functionalities. Clearly links test items to specific requirements or user stories. Addresses different types of testing (e.g., functional, security, performance).	Shows a reasonable connection to software requirements. Provides some categorization of test items (e.g., functional, non-functional).	not comprehensive. Limited clarity on how these items relate to the overall software system.	features will be tested. Lacks connection to the software requirements or functionality.
Test Cases ***40%***	Exceptionally well-documented and comprehensive test cases. Covers a variety of scenarios, including positive, negative, boundary, and edge cases. Fully aligned with user stories and acceptance criteria. Uses structured test case formats (e.g., Given-	Well-structured test cases with detailed steps, expected results, and clear pass/fail criteria. Covers positive, negative, and edge test cases for user stories. Shows strong alignment with acceptance criteria of user stories.	Clearly defined test cases linked to user stories. Includes test steps, expected results, and pass/fail criteria. Covers positive test cases but may lack edge cases or negative scenarios.	Basic test cases are provided but lack detail. Some connection to user stories, but not well-defined. Test cases may not be structured properly (e.g., missing test steps or expected results).	Test cases are missing, unclear, or irrelevant. No link to user stories or software requirements. Lacks test steps, expected results, or pass/fail criteria.

	When-Then for BDD or step-wise execution). Demonstrates critical thinking in test design, ensuring robust validation of user stories.				
Test Results ***10%***	Exceptionally detailed and well-documented test results. Clearly presents pass/fail status, execution timestamps, and environment details. Includes defect tracking, with severity levels and resolution status.	Well-structured test results with clear pass/fail status. Includes test execution logs, defect reports, and summaries of issues found. Provides insights into failed test cases and possible causes.	Clearly recorded test results, including test case execution status. Includes pass/fail status for test cases. Mentions defects found, but without detailed analysis.	Basic test results are provided but lack detail. Some test case outcomes are recorded but may not be well-structured. Limited explanation of pass/fail criteria or issue tracking.	Test results are missing, unclear, or irrelevant. No documentation of executed test cases. No pass/fail status or evidence of testing.

The following Subject Learning Outcomes are addressed in this assessment	
SLO b)	Evaluate different software processes, and assess their suitability in a given context.
SLO c)	Identify, analyse, and manage software project risks.
SLO e)	Plan an effective way of communication between the different stake holders.