

## ASSESSMENT 1 - Part A BRIEF

<b>Subject Code and Title</b>	SEP401 Software Engineering Principles
<b>Assessment</b>	Proposal
<b>Individual/Group</b>	Individual/Collaborative
<b>Length</b>	Maximum two pages, 1000 words (+/- 10%)
<b>Learning Outcomes</b>	This assessment addresses the Subject Learning Outcomes outlined at the bottom of this document.
<b>Submission</b>	Due by 11:55pm AEDT Sunday of Module 3 (Week 3).
<b>Weighting</b>	10%
<b>Total Marks</b>	100 marks

### Context

A software engineering proposal is a document that a software developer submits to a business stakeholder for acceptance. A software project proposal describes what you are going to do, how you're going to do it, and what the results of that process will be. Your proposal should also show the benefits, funding and resources required before you get the approval from your stakeholders. In software development, the project does not exist until it has been proposed and given the approval. The project proposal is your first step at coordinating all the elements of your potential project. This is your first step in managing all the tasks, equipment and other materials you'll need and align them with a feasible schedule to achieve the project's objective.

This assessment will develop your ability to identify a problem, propose how you want to solve it and communicate these ideas to different stakeholders.

All assessments for SEP104 are integrated to help you develop your skills in producing high quality software solutions from the initial software engineering process to delivery. Your project proposal will provide the **goals**, **objectives** and **structure** of what the project looks like to help you focus when identifying and describing requirements for the other assessments. When making design and implementation decisions, you always look at the project proposal to help set the direction of your decisions.

The proposal you create for this first assessment, once approved, will be used for your requirements document that you will deliver in Part B of this assessment. Assessment 2 requires you to develop the Software Design Document and to include any feedback you receive from Part A & B of Assessment 1. Your Software Design Document that you

produced in Assessment 2 should be used as the foundation of the Software Application you are asked to develop in Assessment 3.

## Instructions

This assessment requires you to write and submit a Software project proposal.

Keep in mind that this is a proposal and that your facilitator has to approve it. Modifications and changes to the proposal might be required. The key is to diagnose the problem and propose a solution to convince the customer (reader) to accept your proposal. In this course, you are not required to come up with an innovative solution to an existing problem or a unique problem to solve. Some key steps to help you find a project topic:

- Identify a problem you want to solve (e.g, it takes too long to order food in a restaurant, patient records are still entered manually for some GP clinics, or board games should be accessible using other media)
- Think of possible solutions to solve the problem using software (e.g., create stand-alone app, information systems, online app, mobile app, or extend an existing software) and think of how you plan to achieve this.

*Note: At the end of the subject, you will present an implemented software application of your proposed project. It is important to keep in mind that this is a 12-week project (4 weeks will be allocated to identifying project topic and requirements, 4 weeks will be on designing the project and 4 weeks will be allocated to implementing your project).*

A good guide on defining the scope of your project is that it should have 4-6 major functions/features. The examples below will give you an idea of the level of complexity expected for your project.

### Here are examples of possible projects (you can come up with your own idea):

- 1) Games (single or multi-player, no Artificial Intelligence required)
  - a) Board Games (Reversi, Connect Four, Checkers, Battleship, Scrabble, etc...)
  - b) Card Games (Blackjack, Poker, Solitaire, Uno etc...)
- 2) An application prototype where a user can add/edit/view records. There is no need to save data to files (use an in-memory database). Examples:
  - a) Sports results, fixtures, statistics, players ( for soccer, AFL, Rugby, Cricket, Basketball, Tennis or any sports you prefer)
  - b) Online banking – deposit, withdraw or view accounts
  - c) Food ordering – view menu items from restaurants; add, edit, delete orders

### Document Proposal Format:

Your document should have the following format:

1. Title of the Proposed Project
2. Problem Description
  - Describe the problem that you're planning to solve.
  - Start a brief description of a high-level problem (few sentences) then describe some specific issues that you're interested in, then provide more specific details about the sub-issues that your work will solve.
3. Proposed Solution
  - Describe how do you propose to address the identified problems
  - Discuss the business value of your proposed solution. *What will your customer and users gain from your proposed system?*
4. Project Plan
  - Discuss here the Software Process that you will use for your proposed project.
  - Explain the activities in each of the stages (with the estimated timeline).
  - Explain why this is the appropriate process for your proposed project

### Submission Instructions

1. Submit your document in pdf format in Turnitin via the Assessment link in the main navigation menu in SEP104 Software Engineering Principles. The Learning Facilitator will provide feedback via the Grade Centre in the LMS portal. Feedback can be viewed in My Grades.

## Assessment Rubric

Assessment Attributes	Fail (Unacceptable) 0-49%	Pass (Functional) 50-64%	Credit (Proficient) 65-74%	Distinction (Advanced) 75-84%	High Distinction (Exceptional) 85-100%
<b>Composition</b>  <b>10%</b>	<p>The document lacked overall organisation. Limited underlying logic and flow of ideas overall</p> <p>Grammatical and spelling errors throughout document.</p>	<p>The document was organised and clearly written for most part. Limited logic and/or flow of ideas in some parts of the document.</p> <p>Grammatical and spelling errors were present in some parts of the document.</p>	<p>The document was organised and clearly written. The logic and/or flow of ideas in the entire document were easy to follow in most parts.</p> <p>Vocabulary was well chosen with some minor improvement needed.</p> <p>Sentences were mostly grammatically correct and/or only a few spelling errors were present.</p>	<p>The document was well organised and clearly written. The underlying logic and/or flow of ideas were easy to follow throughout the document</p> <p>Vocabulary was well chosen and supported reader comprehension.</p> <p>Sentences were grammatically correct and free from spelling errors.</p>	<p>The document was well organised and clearly written. The underlying logic was clearly articulated and easy to follow throughout the document</p> <p>Vocabulary was chosen that precisely expressed the intended meaning and supported reader comprehension.</p> <p>Sentences were grammatically correct and free from spelling errors.</p>
<b>High-level Problem Description and Detailed Problem Description</b>  <b>30%</b>	<p>High -level context description of the problem domain is missing or is unclear.</p> <p>Demonstrated little to no ability to define a manageable and achievable topic.</p>	<p>High -level context description of the problem domain is missing or is unclear.</p> <p>Defines a problem that is manageable and achievable, however too narrowly focused and leaves out relevant aspects of the problem; or the problem is so broad that it cannot be</p>	<p>High - level context description of the problem domain is clear.</p> <p>Defined a focused and manageable/achievable problem that appropriately addresses most of the relevant aspects of the problem.</p>	<p>High - level context description of the problem domain is clear and describes the exact purpose of the project.</p> <p>Defined a focused and manageable/achievable problem that appropriately addresses relevant aspects of the problem.</p>	<p>High - level context description of the problem domain is clear, concise and describes the exact purpose of the project.</p> <p>Identified a creative, focused, and manageable problem that addresses potentially significant yet previously less-explored aspects of the problem.</p>

	Key issues and/or problems were not identified; or key issues and/or problems were identified inaccurately. Meaning is unclear.	adequately addressed.  Identified some key issues and/or problems. Descriptions have some inaccuracies or omissions that interfere with the interpretation of the meaning being conveyed.	Identified most key issues and/or problems. Some minor inaccuracies or omissions may be present, but do not interfere with meaning.	Identified all possible key issues and/or problems.	Identified all possible key issues and/or problems and described them clearly, accurately, and appropriately.
<b><i>Proposed Solution</i></b>  <b>30%</b>	Demonstrated little to no ability to identify an approach for solving the problem and/or proposed a solution related to the problem.	Identified only a single approach for solving the problem that does not apply within a specific context in the problem.  Proposed one solution that is “generic” (off the shelf) rather than individually designed to address the specific contextual factors of the problem.	Identified multiple approaches for solving the problem, only some of which apply within a specific context in the problem.  Proposed one or more solutions that indicate some understanding of the problem.	Identified multiple approaches for solving the problem that apply within a specific context of the problem.  Proposed one or more solutions that indicate comprehension of the problem.	Identified multiple and creative approaches for solving the problem that apply within a specific context of the problem.  Proposed one or more solutions that indicate a deep comprehension of problem.
<b><i>Business Value of the proposed solution</i></b>  <b>20%</b>	Did not present the business value (significance) of the proposed solution.  Significance of the proposed solution failed to present analysis and/or present an inaccurate or inappropriate information	Presented some appropriate business value (significance) of the proposed solution. Some relevant information was missing.  Significance of the proposed solution showed limited analysis. May contain inaccuracies or omissions that interfere with analysis and/or meaning.	Presented appropriate business value (significance) of the proposed solution.  Generally showed analysis of significance of the proposed solution. Minor inaccuracies or omissions do not interfere with analysis or meaning.	Presented sufficient and appropriate business value (significance) of the proposed solution.  Showed analysis of significance of the proposed solution.	Presented appropriate, sufficient and credible business value (significance) of the proposed solution.  Significance of the proposed solution showed clear analysis, relevance, and validity.

<b>Project Plan</b>  <b>10%</b>	Software process is not identified or poorly described.	Software process were identified and described.	Identifies appropriate software process for the project.	Competently identifies appropriate software process for the project.	Proficiently identifies appropriate software process for the project.
		Stages of the software process have some description and timeline presented.	Describes the stages of the software process correctly and timeline presented is achievable.	Stages of the software process are presented in detail.	Exceptional description of the stages of the software process.
		Provides justification of the use of the software process.	Justification of the identified software process is described clearly and demonstrate understanding of the software process.	Timeline shows careful planning.	Timeline is thorough and shows that careful planning was applied.
				Justification were clearly described and demonstrate clear understanding of the software process.	Justification identifies all relevant aspects of the project and demonstrates higher level of understanding of the software process.

The following Subject Learning Outcomes are addressed in this assessment	
<b>SLO b)</b>	Author documents required for the software development process e.g.: formal specifications, requirements document, test plan.