ASSESSMENT 2 - BRIEF	
Subject Code and Title	SDM404 Software Development Management
Assessment	Project Requirements and Prototype
Individual/Group	Group
Length	Software Requirements Specification (12-15 pages) Graphical User Interface/ Prototype of the project application (screenshots) Individual contribution report (500 words +/- 10%)
Learning Outcomes	b) Evaluate modern software project management practices to determine the most suitable methodologies for various software development scenarios. d) Develop and justify comprehensive estimates for software project effort, cost and scheduling to support effective decision making and resource optimisation. f) Develop management strategies that address challenges in software projects, emphasising leadership and team coordination skills.
Submission	Due by 11:55pm AEST/AEDT Sunday end of Week 7.
Weighting	30%
Total Marks	100

Assessment Task

Build upon your Assessment 1 Project Proposal and implementation Plan, this assessment requires you to develop a Software Requirements Specification (SRS) document that clearly outlines all the requirements of your proposed software project from a business stakeholder's perspective.

Please ensure that you incorporate any feedback you were provided from Assessment 1.

Context

In software development, it is important to describe how a product will work entirely from the business stakeholders' view. A Software Requirements Specification (SRS) is used to describe all the capabilities a product must have in order to fulfil the business, stakeholder and user needs. Besides establishing a clear agreement between the software developer and customer on what the completed software must do, the SRS serves as a reference for testing and to address the needs of the operations and maintenance teams.

This assessment will allow you to demonstrate your understanding of software requirements analysis and modelling and help develop your business analysis skills as part of the software development team.

Task Instructions

1. Follow the headings and instructions provided in the **Software Requirements Specification (SRS) Template** (attached in the assessment area) to develop a detailed SRS for your proposed system.
2. You are required to complete all section in the SRS and the clearly define the system's inputs, outputs, functions, and attributes, as well as the characteristics of the system environment.
3. You can use any drawing tool in creating your requirements modelling diagrams.
4. Each student must submit an **individual contribution report** outlining their involvement in the development of the software project plan. In this report, you should:
   1) List **the specific tasks** and sections you contributed to.
   2) Describe any **challenges or difficulties** you encountered during the process.
   3) Explain the **solutions you explored and implemented** to address these challenges.
   4) Reflect briefly on your **learning experience** and any skills or insights gained through your contribution.
5. Your report should be approximately **500 words** and demonstrate both accountability and critical reflection on your role within the team.

Referencing

It is essential that you use current APA style for citing and referencing the sources that you use. Please see more information on citing and referencing guidelines on the Academic Success webpage.

Assessment Support

For a range of additional resources and support to help you complete your assessment, please consult the Study Support page on the Student Hub.

Academic Integrity

All students are responsible for ensuring that their submitted work is original, adheres to academic writing standards outlined in the Torrens University Academic Writing Guide, and is appropriately referenced according to the guidelines provided in the Torrens University APA Referencing Guide.

Students need to have read and be aware of the Torrens University Australia Academic Integrity Policy, Academic Integrity Procedure and subsequent penalties for academic misconduct. For more information, please refer to the Academic Integrity guidelines and the Torrens University Library.

Students must also keep all required evidence in making an assessment; a copy of all submitted material and any assessment drafts.

Generative AI

Please refer to the Torrens University Library for guidance on the use of Generative AI. Please speak to your learning facilitator regarding the use of GenAI tools in your assessments.

Submission Instructions

Submit this task via Assessments > Briefs & Submissions in the main navigation menu in SDM404: Software Development Management. Please name your file using the following format:

- SubjectCode_Surname_FirstNameInitial_AssessmentNumber

  e.g. SDM404_Jones_S_Assessment_2

Your marked assessment can be viewed in MyLearn.

Assessment Due Dates and Late Penalties

Assessments may be submitted on or before the due date. Late penalties apply for assessments that are submitted after the due date.
Refer to:

- Assessment Policy for Higher Education Coursework (HE) and ELICOS
  Torrens University | Think Education
- Assessment Special Consideration Guidelines for Students (HE Coursework)
  Torrens University | Think Education
- Student Hub for Assessment Extension Information.

Special Consideration

To apply for special consideration for a modification to an assessment task or exam due to unexpected or extenuating circumstances, please consult the Assessment Policy for Higher Education Coursework and ELICOS and, if applicable to your circumstance, submit a completed Application for Assessment Special Consideration Form to your learning facilitator.

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%**
Overall Description **10 %**	The overall perspective and product functions are fully and clearly defined with comprehensive detail. Stakeholder table includes all relevant stakeholders, with complete and accurate levels of interests and influence for each. Operating Environment is defined logically, thoroughly, and with full contextual relevance. Project Constraints are fully defined, clearly explained, and justified. Assumptions about the project are all realistic, relevant, and well-supported.	The overall perspective and product functions are clearly defined with sufficient detail. Stakeholder table includes all key stakeholders, with clearly stated levels of interests and influence. Operating Environment is defined logically with appropriate contextual detail. Project Constraints are defined well and include reasonable justification. Assumptions about the project are mostly realistic and supported with some explanation.	The overall perspective and product functions are defined clearly. Stakeholder table shows all of the stakeholders along with their level of interests and level of influence. Operating Environment is defined logically. Project Constraints are partially defined and somewhat explained. Assumptions about the project are present but require better justification or realism.	The overall perspective and product functions are defined, but lack clarity or are incomplete. Stakeholder table includes limited information; some stakeholders, interests, or influence levels are missing. Operating Environment is defined but lacks logical structure or necessary detail. Project Constraints are mentioned but are unclear or lack explanation. Assumptions about the project are present but unrealistic or unsupported.	The overall perspective and product functions are poorly defined. Stakeholder table lacks the information. Operating Environment is not defined or is illogical. Project Constraints are missing or not defined well. Assumptions about the project are missing or not realistic.

User Stories **15 %**	The product backlog is well organised and includes user stories that are correctly numbered to uniquely identify each one. All user stories are written in the proper format ("As a..., I want..., so that..."), are prioritised properly, and are assigned story points using the Fibonacci sequence consistently and logically.	The product backlog is well organised, and all user stories are numbered to uniquely identify them. The stories are written in the correct format and prioritised properly, with only minor inconsistencies. Story points are assigned and mostly follow the Fibonacci sequence, though some inconsistencies may exist.	The product backlog is organised, and user stories are uniquely numbered. Most user stories follow the correct format, though a few may contain small errors. Prioritisation is applied but could be more consistent. Story points are assigned, though they may occasionally lack consistency with the Fibonacci scale.	The product backlog shows some organisation, but it lacks consistency. User stories may be partially numbered or inconsistently formatted. Prioritisation is weak or unclear, and story points are assigned without clear logic or method.	The product backlog is poorly organised. User stories are not properly numbered, not written in the correct format, not prioritised properly, and story points are either not assigned or illogical.
Use Case Diagram + Sue Case Narratives **35 %**	The use case diagram is nicely drawn, accurately showing all use cases, actors, and their interactions. Use case narratives are clearly related to user stories through proper numbering. Each narrative includes well-defined actors, pre-conditions, post-conditions, and triggers. The "Main Scenario" includes a detailed and logical flow of steps, and both "Alternate Scenarios"	The use case diagram is nicely drawn and includes all relevant use cases, actors, and their interactions. Use case narratives are linked to user stories with proper numbering and include appropriate actors, pre-conditions, post-conditions, and triggers. The "Main Scenario" is described with a clear and logical flow of steps, and both "Alternate Scenarios" and "Exceptional Scenarios" are logically presented and described.	The use case diagram is reasonably drawn and shows all required use cases, actors, and interactions. Use case narratives are related to user stories with proper numbering and include the correct structure with actors, pre-conditions, post-conditions, and triggers. The "Main Scenario", "Alternate Scenarios" and "Exceptional Scenarios" are detailed and mostly logical.	The use case diagram is present but may be incomplete or inconsistently drawn, missing some use cases, actors, or their interactions. Use case narratives are related to user stories, but numbering or structure may be inconsistent. Actors, pre-conditions, post-conditions, or triggers are mentioned but are not properly defined. The "Main Scenario" lacks clarity and structure, and "Alternate" or "Exceptional	The use case diagram is poorly drawn, missing key use cases, actors, and their interactions. Use case narratives are not properly related to user stories and lack correct numbering. Narratives do not include proper actors, pre-conditions, post-conditions, or triggers. The "Main Scenario" lacks detail and logical flow, while "Alternate Scenarios" and "Exceptional Scenarios" are

	and "Exceptional Scenarios" follow a proper and logical structure. Multiple approaches for solving the problem are identified, and proposed solutions are specifically tailored to the contextual factors of the problem.	Multiple relevant approaches are considered, and the proposed solutions reflect a solid understanding of the problem context.	A single approach is identified, and the solution shows some understanding, but it remains generic rather than context-specific.	Scenarios" are not logically presented. Only one generic approach is identified, and the proposed solution is not tailored to the problem context.	either missing or not logically structured. No suitable solution is proposed, or only a generic approach is provided without considering the specific context of the problem.
Non-Functional requirements **15%**	Non-functional requirements are clearly written in an objective manner and are fully measurable. Performance requirements are quantified using precise metrics and can be easily measured. Safety requirements are fully relevant and appropriately integrated into the system. Security requirements are clearly measurable and directly tied to the functionality of the system. The business value of the proposed solution is comprehensively presented, with well-	Non-functional requirements are written objectively and are mostly measurable, though some could benefit from greater specificity. Performance requirements are quantified and measurable, but may lack full detail or clarity in some aspects. Safety requirements are relevant to the system but may not be fully explored or integrated into the design. Security requirements are measurable and linked to functionality, though some minor areas of integration or clarity may be lacking. The business value of the proposed solution is clearly presented with strong	Non-functional requirements are written objectively and mostly measurable, but some requirements may lack clarity or precision. Performance requirements are quantifiable but may not be fully specified or consistent across the system. Safety requirements may not be fully relevant or integrated into the system design, but they are acknowledged. Security requirements are weakly measurable or only partially related to the system's functionality. The business value of the proposed solution is	Non-functional requirements are written objectively but are generally vague or not entirely measurable. Performance requirements are not clearly quantified or lack measurement criteria. Safety requirements are identified but may not be relevant or considered in the system design. Security requirements are neither measurable nor directly related to system functionality. The business value of the proposed solution is poorly presented, with minimal analysis, and significant inaccuracies or omissions that reduce its credibility.	Non-functional requirements are written subjectively and are not measurable. Performance requirements are not quantified, and it is unclear how they can be measured. Safety requirements are not relevant to the system and are not considered. Security requirements are neither measurable nor related to the system’s functionality. The business value of the proposed solution is not presented, or the analysis is entirely missing, incorrect, or inappropriate.

	supported analysis demonstrating its clear relevance, impact, and validity.	analysis, but may contain minor inaccuracies or omitting some supporting evidence.	presented with some relevant information, but the analysis is underdeveloped and may contain notable inaccuracies or gaps.		
External Interface Requirements ***5%***	All external, software, and hardware interfaces are clearly mentioned and are required for the system. The mechanism of each interface is clearly defined, showing how interaction occurs between system components or external systems.	All external interfaces are mentioned and are required for the system. Both software and hardware interfaces are listed and relevant, but the mechanism of interaction is not fully defined or lacks detail.	All external interfaces are mentioned. Software interfaces listed are required for the system, but hardware interfaces are not relevant or are inaccurately included, or vice versa. The mechanism of interfaces is mentioned but not clearly defined.	All external interfaces are mentioned, but their relevance to the system is unclear. Either software or hardware interfaces are not required for the system. The mechanism of interfaces is vague or missing key details.	External interfaces are not clearly mentioned. Software and hardware interfaces listed are not required for the system. The mechanism of interfaces is not defined.
GUI/ Prototype Screens ***30%***	Prototype screens are fully aligned with the system's functionality and use cases. The design is highly polished and professional, with consistent use of colour, typography, layout, and spacing. The prototype is fully functional, with smooth interactions, intuitive navigation, and clear affordances.	Prototype screens are relevant to the system's functionality and use cases. The interface is clean and consistent overall, with only minor issues in visual design or layout. The prototype is functional with mostly smooth interactions and effective navigation. Usability is strong, though some areas could benefit from slight refinement.	Prototype screens generally reflect the system's functionality and use cases. The design is visually acceptable but lacks polish and has inconsistencies in layout or design elements. The prototype is mostly functional, but some interactions are incomplete or unclear. Navigation is usable but may cause some confusion.	The prototype has basic visual structure but includes major inconsistencies in layout, colour, or typography. Functionality is limited, with broken or missing interactions and unclear navigation. Usability is compromised by design flaws or poor flow. Design choices are generic and do not reflect user-centered thinking.	Prototype screens are misaligned with system functionality or missing entirely. The design is cluttered, inconsistent, or confusing. The prototype is non-functional or very difficult to use. Design decisions are incoherent, with no evidence of usability or visual planning.

	Accessibility and usability are well-considered throughout the design. Design choices are original and enhance both user experience and system clarity.	Design decisions are thoughtful, though not especially unique.	The design meets expectations but lacks originality and attention to detail.		