OCCUPATIONAL CERTIFICATE: SOFTWARE ENGINEER

Formative Assessment 1

Software Design Engineering

SDE631

2025

CTU training solutions | Private Tertiary Education

Please complete the declaration of authenticity below for all assignments:

DECLARATION OF AUTHENTICITY

I ____

(FULL NAME)

hereby declare that the contents of this assignment are entirely my own work with the exception of the following elements: (List the elements of work in this project that were not self-generated as well as who the originator of the element is)

Element	Originator

Signature: ______________________________

Date: _____________

Occupational Certificate: Software Engineer

Module: Software Design and Development

Module Code: SDE631

NQF: 6

Credits: 75

Assessment type: Formative Assessment

Mark allocation: 100

Hand out date: 07/03/2025

Hand in date: 20/03/2025

Instructions:

Read each question carefully and consider the mark allocation prior to answering.

- Ensure you answer all the questions.
- For final submission**,** include the Declaration of Authenticity and your Answer Sheet (PDF Document)

Exit Level Outcomes:

1. Apply knowledge to design software to meet clients' needs.
2. Design and manipulate databases.
3. Develop software to add value to the organization.
4. Test or debug source code.

Assessment Outcomes:

Upon completion of this assessment, students should be able to:

- Understand the Fundamentals of System Analysis and Design
- Apply System Planning Techniques
- Perform Comprehensive System Analysis
- Design Effective Systems
- Develop and Maintain System Documentation
- Perform Problem Identification and Definition
- Engage with Stakeholders
- Collaborate with Team Members
- Develop and Execute a Project Plan
- Demonstrate Research and Critical Thinking Skills

Question 1 (50 Marks)

Instructions: Read the scenario below and complete the following tasks

1.1 Overview/Introduction to System Analysis and Design (20)

a) Define the Systems Development Life Cycle (SDLC) and outline its key stages. (4)

b) Identify the SDLC phase in which functional and non-functional requirements are defined and justify its significance in system development. (3)

c) State the SDLC phase in which version control is utilized and provide an example of a version control tool. (3)

d) Analyze the key challenges encountered in SDLC and evaluate their impact on project success. (10)

1.2 System Planning (15)

a) Explain the role of Data Flow Diagrams (DFDs) and Entity-Relationship Diagrams (ERDs) in system analysis, highlighting their importance in the planning phase. (5)

b) Illustrate how DFDs and ERDs can be applied to model a real-world information system by providing a detailed example. (5)

c) Evaluate the role of comprehensive system documentation in the onboarding process for new team members, emphasizing its benefits and challenges. (5)

1.3 System Analysis (15)

a) Discuss the challenges faced in designing user interfaces and user experiences (UI/UX) for complex systems and propose possible solutions to overcome these challenges. (5)

b) Examine how system designers can ensure that the final design meets both functional and non-functional requirements, providing practical examples. (5)

c) List and describe key tools used in various phases of the Software Development Life Cycle (SDLC), and assess their effectiveness in improving system development. (5)

Question 2 (25 Marks)

Instructions: Read the scenario below and complete the following tasks, where appropriate use hypothetical actors.

Scenario – Managing Legacy Systems (25 Marks)

A company needs to integrate new features into legacy systems while maintaining stability and performance.

a) Propose a strategy for updating a legacy system while operating within the constraints of its existing SDLC model. Explain how backward compatibility can be maintained during this process. (5)

b) Analyze the challenges related to documentation, testing, and deployment when dealing with legacy code. Recommend strategies to mitigate these challenges. (10)

c) Evaluate how modern SDLC practices, such as microservices architecture, can be leveraged to incrementally modernize parts of the system without requiring a complete rewrite. Justify your approach with practical examples. (10)

Question 3 (25 Marks)

Instructions: Read the scenario below and complete the following tasks, where appropriate use hypothetical actors.

Scenario - Multi-Platform Development

A software product needs to be developed for both web and mobile platforms simultaneously.

a) Design an SDLC structure that supports parallel development for multiple platforms while ensuring feature parity and code reuse. (8)

b) Recommend tools and methodologies that can be used to maintain a consistent user experience across both web and mobile platforms. Explain how these tools facilitate seamless cross-platform development. (7)

c) Assess the implications of this multi-platform approach on the testing phase. Distinguish between platform-specific testing and shared functionality testing, and propose best practices for handling both. (10)

End of Questions **[Total = 100 Marks]**

End of paper.

Marking rubric

Question 1: System Analysis and Design (50 Marks)

1.1 Overview/Introduction to System Analysis and Design (20 Marks)

Criteria	Poor (0 – 3 Marks)	Needs Improvement (4 – 7 Marks)	Excellent (8 – 10 Marks)
Understanding of SDLC & Key Stages (4 Marks)	Inaccurate or incomplete definition, missing key SDLC stages.	Basic but incomplete definition; some SDLC stages included but lacking depth.	Clear and precise definition with all SDLC stages accurately outlined.
SDLC Phase for Defining Requirements & Justification (3 Marks)	Incorrect phase identified with no justification.	Correct phase identified, but justification is unclear or weak.	Correct phase identified with a well-supported justification.
SDLC Phase for Version Control & Example (3 Marks)	Incorrect or missing phase and tool.	Correct phase identified, but tool example is unclear or inappropriate.	Correct phase and appropriate version control tool provided.
Challenges in SDLC & Impact on Project Success (10 Marks)	Challenges poorly identified with no analysis of impact.	Challenges identified but analysis lacks depth or clarity.	Challenges well analyzed with strong justification of impact on project success.

1.2 System Planning (15 Marks)

Criteria	Poor (0 – 5 Marks)	Needs Improvement (6 – 10 Marks)	Excellent (11 – 15 Marks)
DFDs & ERDs in System Planning (5 Marks)	Incorrect or unclear explanation; missing importance.	Basic understanding with partial explanation of importance.	Clear and comprehensive explanation of DFDs and ERDs with their significance.
Application of DFDs & ERDs with Example (5 Marks)	No or vague example; unclear modeling.	Example provided but lacks details or clarity.	Well-structured example with detailed application.
Importance of System Documentation (5 Marks)	Poor explanation with no discussion of benefits/challenges.	Some discussion but lacks depth or clarity.	Well-evaluated discussion with clear benefits and challenges.

1.3 System Analysis (15 Marks)

Criteria	Poor (0 – 5 Marks)	Needs Improvement (6 – 10 Marks)	Excellent (11 – 15 Marks)
UI/UX Challenges & Solutions (5 Marks)	Incorrect or minimal discussion with no solutions.	Some challenges discussed, but solutions are vague.	Well-structured discussion with strong solutions.
Meeting Functional & Non-Functional Requirements (5 Marks)	Poor or unclear explanation with no practical examples.	Some practical examples, but lacking in depth.	Strong examples provided with clear justifications.
Tools in SDLC & Effectiveness (5 Marks)	Incorrect tools listed; no discussion of effectiveness.	Some tools identified, but evaluation is weak.	Well-explained tools with a strong assessment of their effectiveness.

Question 2: Managing Legacy Systems **(25 Marks)**

Criteria	Poor (0 – 5 Marks)	Needs Improvement (6 – 10 Marks)	Excellent (11 – 15 Marks)
Strategy for Updating Legacy Systems (5 Marks)	Strategy is unclear or impractical; no mention of backward compatibility.	Some useful strategies discussed, but lacks clarity or detail.	Strong and feasible strategy with clear backward compatibility solutions.
Challenges in Documentation, Testing, Deployment (10 Marks)	Challenges are vague or poorly analyzed.	Challenges identified, but solutions lack depth.	Well-analyzed challenges with practical mitigation strategies.
Modern SDLC Practices for Legacy Modernization (10 Marks)	Minimal or incorrect discussion of microservices; lacks justification.	Some discussion on modernization, but weak justification.	Strong evaluation of microservices with well-supported examples.

Question 3: Multi-Platform Development **(25 Marks)**

Criteria	Poor (0 – 5 Marks)	Needs Improvement (6 – 10 Marks)	Excellent (11 – 15 Marks)
SDLC for Parallel Development (8 Marks)	Poor or unclear structure for parallel development.	Some structure provided but lacks key considerations.	Well-structured SDLC approach ensuring feature parity and reuse.
Tools & Methodologies for Consistency (7 Marks)	Incorrect or irrelevant tools listed.	Some useful tools identified, but lacking explanation.	Strong recommendations with clear justifications.
Testing Implications & Best Practices (10 Marks)	Poor analysis with unclear best practices.	Some testing considerations provided, but lacks structure.	Strong evaluation with detailed best practices.