Course: Software Requirements Engineering

Week 16: Final Exam

Lecturer: Yimer Amedie (MSc.)
Addis Ababa Science and Technology University, Ethiopia

June, 2026

Instruction

Before you begin the final exam, please take a moment to review these important instructions:

- ✓ Read each question carefully before answering.
- ✓ The exam includes a mix of multiple choice, short answer, and case-based questions.
- ✓ Work independently
- ✓ Manage your time wisely

Exam Structure

- Multiple Choice Questions (30 questions)
- Short Answer Questions (5 questions)
- Case Analysis Questions (3 questions)

  - Time allotted: **90 minutes**
  - Total point: **100%**

Questions

Part I: Multiple Choice

Instruction: Choose the correct answer from the given alternatives (2 pts each)

Questions (1-2/30)

1. Which of the following best describes Software Requirements Engineering?

a. Software coding process

b. Process of understanding, documenting, and managing requirements

c. Database administration

d. Testing activity

2. A functional requirement specifies:

a. Performance level

b. Security quality

c. What the system should do

d. Project budget

Questions (3-4/30)

3. Which requirement is an example of non-functional requirement?

a. The system shall generate reports

b. The system shall allow login

c. The application shall respond within 2 seconds

d. The system shall export PDF

4. Which elicitation technique is most appropriate for understanding hidden work practices?

a. Interview

b. Questionnaire

c. Observation

d. Brainstorming

Questions (5-6/30)

5. Stakeholders are:

a. Only software developers

b. Individuals affected by the system

c. Only customers

d. Only testers

6. Requirement ambiguity leads to:

a. Improved software quality

b. Reduced misunderstanding

c. Misinterpretation of system needs

d. Faster deployment

Questions (7-8/30)

7. Which document formally records software requirements?

a. Test Plan

b. SRS

c. User Manual

d. Deployment Plan

8. The MoSCoW technique is used for:

a. Testing

b. Requirement prioritization

c. Coding

d. Risk management

Questions (9-10/30)

9. Which UML diagram mainly represents interactions between user and system?

a. Activity Diagram

b. Sequence Diagram

c. Use Case Diagram

d. Class Diagram

10. Requirement validation ensures:

a. Correct code exists

b. Right requirements are identified

c. Database performance

d. UI consistency only

Questions (11-12/30)

11. Requirement validation ensures that:

a. The software code is optimized

b. The right requirements are defined

c. Testing is completed

d. Hardware is installed

12. Requirement verification checks whether:

a. Requirements are written correctly and follow standards

b. Stakeholders are removed

c. Developers finish coding

d. Servers are configured

Questions (13-14/30)

13. Which technique is commonly used for requirement validation?

a. Requirement review

b. Hardware testing

c. Coding standards

d. Data backup

14. Requirement inspections are conducted to:

a. Find defects in requirements documents

b. Replace system testing

c. Write code automatically

d. Install software

Questions (15-16/30)

15. Prototyping helps validation by:

a. Ignoring stakeholder feedback

b. Allowing stakeholders to visualize system behavior

c. Replacing requirement gathering

d. Eliminating system analysis

16. Requirement traceability refers to:

a. Tracking requirements throughout the project lifecycle

b. Software coding process

c. Hardware maintenance

d. Team scheduling

Questions (17-18/30)

17. A traceability matrix is mainly used to:

a. Store passwords

b. Map requirements to design, development, and testing

c. Replace requirement documents

d. Conduct interviews

18. Forward traceability ensures:

a. Every requirement is implemented

b. Code is deleted

c. Stakeholders are ignored

d. Budgets are approved

Questions (19-20/30)

19. Backward traceability ensures:

a. Every implemented feature maps to a requirement

b. Only developers manage requirements

c. Testing is unnecessary

d. Deployment is faster

20. Why is traceability important?

a. To increase ambiguity

b. To manage requirement changes and ensure coverage

c. To avoid documentation

d. To eliminate stakeholders

Questions (21-22/30)

21. Which tool is commonly used in Agile teams for backlog and requirement management?

a. Jira

b. Calculator

c. Paint

d. Chrome

22. The major benefit of prototyping is:

a. Increased misunderstanding

b. Early stakeholder feedback and requirement clarification

c. Elimination of requirement validation

d. Removal of documentation

Questions (23-24/30)

23. Agile Requirements Engineering emphasizes:

a. Fixed requirements only

b. Continuous feedback and iterative development

c. No stakeholder involvement

d. Extensive upfront documentation only

24. In Agile, requirements are usually documented as:

a. Technical manuals

b. User stories

c. Database schema

d. Source code

Questions (25-26/30)

25. A user story mainly focuses on:

a. Programming syntax

b. User needs and business value

c. Hardware performance

d. Testing scripts

26. Which is the standard format of a user story?

a. As a [User], I want [Function], so that [Benefit]

b. Build software quickly

c. Write code first, test later

d. Design database before requirements

Questions (27-28/30)

27. Product backlog refers to:

a. List of completed tasks only

b. Prioritized list of project requirements/features

c. Developer attendance sheet

d. Hardware inventory

28. Sprint backlog contains:

a. All organizational policies

b. Tasks selected for the current sprint

c. Final software documentation

d. Customer payment details

Questions (29-30/30)

29. The Product Owner in Agile mainly:

a. Writes all software code

b. Manages and prioritizes backlog items

c. Installs servers

d. Conducts hardware maintenance

30. Agile requirements are generally:

a. Fixed and never change

b. Flexible and continuously refined

c. Ignored during development

d. Approved only after coding

Questions

Part II: Short Answer

Instruction: Give short answer for the following questions (3 pts each).

Questions (1-5/5)

1. Define Software Requirements Engineering.
2. Differentiate between functional and non-functional requirements using examples.
3. Explain three requirement elicitation techniques.
4. Why is stakeholder identification important?
5. Explain the role of Software Requirements Specification (SRS).

Questions

Part III: Case Analysis

Instruction: Analyze the following case scenario and provide your answer.

Questions (1/3)

Case 1: Hospital Management System

A hospital plans to implement an online system for patient registration, appointment scheduling, billing, and electronic medical records. Doctors require quick access to records, while management requires strong security and reporting.

Tasks:

a. Identify **five stakeholders**.

b. Write **three functional requirements**.

c. Write **two non-functional requirements**.

Questions (2/3)

Case 2: Banking Mobile Application

A bank is developing a mobile application. Customers want fast login and quick transactions, while the security department demands multi-factor authentication.

Tasks:

a. Identify the requirement conflict.

b. Suggest a negotiation approach.

c. Prioritize requirements using **MoSCoW**.

Questions (3/3)

Case 3: Agile E-Commerce Project

An Agile team is building an online shopping platform.

Tasks:

a. Write **three user stories**.

b. Prepare a simple **product backlog**.

c. Explain how stakeholder feedback would be handled during sprint reviews.

The End!