SENG 11232 · Engineering Foundation
STUDENT PRE-LECTURE NOTES · LECTURE 07

Systems Thinking & Requirements Engineering

Module 3 — Engineering Process & Design

HOW TO USE Read these notes BEFORE the lecture. Highlight terms you don't recognise. Bring your questions to class. Use the blank spaces for your own notes.

Learning Objectives

By the end of this lecture, you should be able to:

- 1. Apply systems thinking to identify components, boundaries, interfaces and emergent behaviour
- 2. Perform stakeholder analysis to identify requirements sources
- 3. Write functional and non-functional requirements that are SMART
- 4. Model requirements using use cases and user stories
- 5. Apply the ISO/IEC 25010 software quality model

Systems Thinking

Key Concepts

- System boundary: defines what is inside (scope) vs. outside (environment)
- Interface: contract between system and environment (inputs/outputs)
- Emergent behaviour: system properties not predictable from individual parts
- Feedback loop — negative (corrective/stabilising) or positive (amplifying)

KEY Most complex software failures are systems failures, not component failures. Test the system as a whole, not just individual parts.

Your notes:

Requirements Engineering

Functional vs Non-Functional Requirements

- Functional: what the system does — specific behaviours
  - Example: "The system shall allow users to log in with email and password"
- Non-functional: how well it does it — quality attributes
  - Example: "Response time ≤ 2 seconds for 95% of requests under 1000 concurrent users"

SMART Requirements

- Specific: clearly stated, no ambiguity
- Measurable: can be tested — has a number or threshold
- Achievable: technically feasible
- Relevant: traces to a stakeholder need
- Testable: you can write a pass/fail test for it

NOTE	IBM research: fixing a requirements defect costs 1× at requirements stage, 10× at design stage, 100× in production.

User Story Format

As a [user type], I want to [action] so that [benefit].

Acceptance criteria: Given [context], When [action], Then [observable result].

Your notes:

_____

_____

_____

Key Terms for This Lecture

Term	Your Definition (fill in during/after lecture)
System Boundary	
Emergent Behaviour	
Feedback Loop	
Stakeholder	
Functional Requirement	
Non-Functional Requirement	
SMART	
Use Case	
User Story	
Acceptance Criteria	
ISO 25010	

Self-Check Questions

Answer these after the lecture to test your understanding:

- Q1. Identify 3 stakeholders for a hospital patient booking system and their primary requirements.

Answer: _____

- Q2. Write one functional and one non-functional (performance) requirement for an online banking app.

Answer: _____

- Q3. Explain why 'the system shall be user-friendly' is not a valid requirement.

Answer: _____

- Q4. Write a user story with 2 acceptance criteria for a student grade-checking feature.

Answer: _____

SENG 11232 Engineering Foundation · Lecture 07: Systems Thinking & Requirements Engineering