SENG 11232 · Engineering Foundation

STUDENT PRE-LECTURE NOTES · LECTURE 12

Project Engineering & the Software Development Lifecycle

Module 4 — Networks, Security & Professional Practice

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. Compare SDLC models and choose appropriately for a given project
- 2. Apply the triple constraint to analyse project trade-offs
- 3. Use story points and planning poker for estimation
- 4. Build a risk register for a software project
- 5. Describe Agile/Scrum ceremonies and Git branching strategy

SDLC Models & Project Management

Major SDLC Models

- Waterfall: sequential phases; best for stable requirements, regulated industries
- Iterative/Agile: short sprints; best for evolving requirements, consumer software
- V-Model: testing maps to each design level; used in safety-critical systems

Triple Constraint (Iron Triangle)

- Scope, Time, Cost — you can only control two; the third is determined by them
- Quality is always at risk when the triangle is unbalanced

KEY	Agile reframes the triangle: quality and time are fixed (sprint length); scope is the variable. Each sprint delivers a vertical slice of working software.

Your notes:

____

____

____

Estimation, Risk & Agile

Estimation

- Story Points: relative estimates on Fibonacci scale (1,2,3,5,8,13...)
- Planning Poker: team consensus — reveals hidden complexity through discussion
- Cone of Uncertainty: at project start, estimate may be 4× to 0.25× of actual

Risk Management

- Identify → Assess (probability × impact) → Respond → Monitor
- Responses: Avoid | Transfer | Mitigate | Accept

Agile / Scrum

- Roles: Product Owner, Scrum Master, Development Team
- Ceremonies: Sprint Planning, Daily Standup, Sprint Review, Retrospective

Git Branching Strategy

- main (stable), develop, feature/xxx, release/xxx
- Pull Request + code review before merge to protected branches
- CI runs tests automatically on every commit

NOTE	Version control is not optional — it is a professional engineering standard. Every project you work on, however small, should use Git from day one.

Your notes:

Key Terms for This Lecture

Term	Your Definition (fill in during/after lecture)
SDLC	
Waterfall	
Agile	
Scrum	
Triple Constraint	
Scope Creep	
Work Breakdown Structure	
Story Points	
Planning Poker	
Risk Register	
Sprint	
Product Backlog	
Definition of Done	
Git	
CI/CD	

Self-Check Questions

Answer these after the lecture to test your understanding:

- Q1. When would you choose a waterfall model over Agile? Name two project characteristics that favour waterfall.

Answer: ____

- Q2. A project has fixed budget and deadline. A new feature request arrives mid-project. What should you do?

Answer: ____

- Q3. Build a 4-row risk register for a 3-month student software project. Include probability, impact and mitigation.

Answer: ____

- Q4. What is the Definition of Done and why is it important in Scrum?

Answer: ____

SENG 11232 Engineering Foundation · Lecture 12: Project Engineering & the Software Development Lifecycle