

# Software and Security Engineering: Supervision 1

Lectures covered by the supervision:

<https://www.cl.cam.ac.uk/teaching/2526/SWSecEng/>

- Lecture 1: Engineering Mindset & The Cost of Failure Or: Why do we need this course?
- Lecture 2: Requirements and Specifications Or: How do we decide what to build?
- Lecture 3: Attitudes to risk

Past exam questions:

<https://www.cl.cam.ac.uk/teaching/exams/pastpapers/t-SoftwareandSecurityEngineering.html>

Relevant examples : <https://www.cl.cam.ac.uk/teaching/2526/SWSecEng/examples.pdf>

Supervision questions:

1. When making an estimate of software project costs, what are the factors that should be considered? What is the ratio of costs between development and maintenance? Provide reasoning for the given ratio. Discuss Brooks's Law in this context.
2. Why is it hard to adequately define non-functional requirements? Consider this example:
  - a. A startup wants to build a drone for "ultra-fast and safe pizza delivery in urban environments". Critique the requirement "The drone must be ultra-fast and safe". Use the standard criteria for good requirements.
  - b. Propose three testable Non-Functional Requirements (NFRs) for this drone, covering performance, safety, and reliability.
3. Explain the process of "design space exploration" and why is it often necessary to make trade-offs when making decisions about software design.
  - a. In the previous question, we mentioned software for drones. In that context, provide software design decisions regarding 3 mentioned NFRs (performance, safety, and reliability) and justify them to a project commissioner (non-technical investor).
4. Provide examples of technical debt in some of the applications that you have developed.
5. What is UML? Is it formally defined? What are its benefits and drawbacks?
6. Explain the concept of traceability with an example. Why does engineering of some software products (e.g., health care related) requires explicit traceability?
7. Discuss sources of complexity, as well as the role of contextual and domain knowledge when engineering two applications: a personal "To-Do List" app versus a team of 50 building a "National Health Records System".
8. Compare strengths and weaknesses of iterative model and spiral model of development.
9. Compare Waterfall, Agile, and continuous engineering development methodologies. Can you think of an example where Waterfall is more adequate than Agile?
  - a. How does the choice of a development methodology influence business model of a software development company?
10. Summarize the main message from lecture 1 in 1-3 sentences?
11. Summarize the main message from lecture 2 in 1-3 sentences?
12. Summarize the main message from lecture 3 in 1-3 sentences?

Bonus task:

1. As part of the supervisions, I would like to offer a set of slides that will help you to establish a continuous engineering environment. Therefore, as the first bonus task, I suggest that you create a simple, test-driven, software project and commit it to GitHub. Please provide a link to the project.

2. Create a project board in GitHub (basic Kanban recommended) and add few future user stories. Provide the link to the project.  
<https://docs.github.com/en/issues/organizing-your-work-with-project-boards/managing-project-boards/about-project-boards>
3. Specify interfaces between components in your project: <https://app.diagrams.net/>  
Provide the link to the project.

Save your answers into MS Teams or email them to me. Please use the following naming pattern:

SASE\_Supervision\_1\_Answers\_<last name>\_<first name>\_Easter\_2026

Send your answers as a pdf, doc, image, or any other format of a document for which there exists an easily available software to open.

Jasmin JAHIC

jj542@cam.ac.uk

<https://www.cl.cam.ac.uk/~jj542/>

<https://www.cl.cam.ac.uk/~jj542/teaching.html>