

Fachhochschule Graubünden
University of Applied Sciences

AISE502: AI in Software Engineering II

Lecture 14: Synthesis, Presentations, and Architecture Defence

Script: synthesis of Parts I–V (no new material); project exercise sheet, M6

Agenda

1. **Lesson 1 – synthesis and exam hints** (*no new material*): where we are; what did AI change; the script read backwards; **Maxim 8**; one discipline at two binding sites
2. **Exam hints**: the written examination – the facts; six learning objectives, five parts; open book: the map and the four tables; **Q&A**
3. **Lessons 2–4 – final presentations, architecture defence, peer reviews** (deliverable **A3**, milestone **M6**): the format; what to present; defend your trade-offs; reflect on where AI helped and where it hurt – in building (A) and in the system (B); assessment of the project
4. Closing: the written examination as the last milestone

Where we are: the semester at a glance

Weeks	Lectures	Exercise / project	Deliverables
1–3	Part I: the framework – dimensions, scenarios, weights	Kickoff; requirements workshops	**A1** (week 3)
4–7	Part II: seven patterns; the match	Architecture study; the match; design	**A2** + **design gate** (week 7)
8–13	Parts III–V: classes, fit at scale, the AI dimension	Implementation sprints + coaching	Milestones (weeks 9/11/13)
14	**Synthesis, exam preparation**	**Presentations + defence**	**A3**

Design phase (weeks 1–7): 2 lessons lecture + 2 exercise – design first, code later. **Implementation phase** (weeks 8–13): 3 lessons lecture + 1 coaching. **Week 14**: 1 lesson synthesis and exam hints + 3 lessons presentations, architecture defence, and peer reviews – then the written examination.

Lecture 13 was the last lecture of new material – today's first lesson is a recap of its synthesis and exam orientation, plus your questions.

What did AI change? Two temptations, one position

What, in the end, did AI change? – the bet of the opening, settled in Lecture 13. Architecture teaching in the AI era faces **two symmetric temptations**:

Denial

- generative AI as a passing tool fashion that leaves software engineering untouched
- **falsified by the adoption data alone**: **90 %** of practitioners report using AI at work by 2025

Exceptionalism

- AI systems as a new discipline with new vocabulary, new roles, new decision logic
- **rejected on the assembled evidence**: nothing AI does, on either axis, required a decision no ADR can record, a correctness no response measure can capture, or a structure no tactic vocabulary describes

Key Concept

Between the temptations lies the position defended since Part I: **absorption**.

The script read backwards as a single argument: Parts I–V

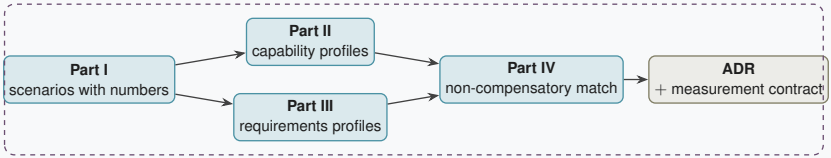
Part	What it established	Artefact
Part I	architecture is the set of significant, hard-to-reverse decisions (**A1**); everything is a trade-off (**A2**); quality attributes – not features – drive structure (**A3**); requirements decide anything only as measurable scenarios (**A4**)	the **twelve dimensions**
Part II	turned seven patterns into **capability profiles** by explaining every rating through the tactics a pattern bundles or impedes	the consolidated **capability table**
Part III	turned ten application classes into **requirements profiles** – recurring bundles of architecturally significant requirements with response measures and hard constraints	the **requirements table**
Part IV	matched them: knock-out screening, veto rule, holistic ordinal reading; **hybrids are the normal case**; the procedure ends in an **ADR**, every ADR in a **measurement contract** – a decision is a hypothesis tested over the life cycle (**A5**)	the 7×10 **fit matrix**
Part V	subjected the whole construction to its hardest contemporary stress test – **and the construction held** (**A6**): Axis A raised the value of the artefacts (documentation as agent context, fitness functions as operating licence); Axis B was absorbed as one class, one dimension, five cell shifts, one composition pattern	Axis A / Axis B outcomes

Maxim 8 – the pipeline of this script in one sentence

Key Concept

Maxim 8. The theory absorbs AI: a tenth application class, a twelfth dimension, shifted cells, one emergent composition pattern – *same assumptions, same procedure, same contract*. The pipeline of this script is one sentence long:

scenarios with numbers (Part I) meet capability profiles (Part II) and requirements profiles (Part III) in a non-compensatory match (Part IV) whose result is an ADR with a measurement contract – and nothing about AI, on either axis, changes a single step of it (Part V)



Part V: nothing about AI, on either axis, changes a single step

The deepest result: one discipline at two binding sites

One symmetry deserves to be made explicit – the deepest result of Part V, and the frame for your reflection in lessons 2–4:

Axis A concluded

- *generation* is cheap
- *verification* is the bottleneck
- so the **SDLC** must be rebuilt around machine-checkable specifications, guardrails, and human accountability

Axis B concluded

- *model output* is cheap
- *validation* is the bottleneck
- so the **runtime** must be rebuilt around contracts, ontology guards, eval harnesses, and a human-owned determinism boundary

Key Concept

These are the **same conclusion at two different binding sites**: whether the LLM sits in the toolchain or in the product, the discipline it demands is **containing cheap, fallible generation behind expensive, explicit verification** – which is, on reflection, what this module has called *architecture* all along.

The written examination – the facts

Written examination (50 %)

- **50 %** of the module grade – the project is the other 50 %
- **End of semester, 60 minutes**
- **Open book:** script and own notes, *on paper*
- **Closed internet**
- Focus: **architecture reasoning – profiles, matching, trade-offs**
- Language: all materials, the script, and the exam in English

Unchanged since Lecture 1 – and Lecture 14 adds no new material; the reading for the examination is the whole script, Parts I–V.

Six learning objectives, five parts

Learning objective (Lecture 1)	Where the script settles it
1. explain why architecture selection is a **matching problem** – no pattern is good or bad in itself	Part I: A2 everything is a trade-off, A3 quality attributes drive structure; Part II: every rating explained through the tactics a pattern bundles or impedes
2. construct a **requirements profile** $R(a)$: scenarios with response measures, utility tree, weights	Part I: A4 requirements decide only as measurable scenarios; Part III: ten classes as ASR bundles with response measures and hard constraints $K(a)$
3. derive the **capability profile** $C(p)$ of seven patterns from their tactics	Part II: capability profiles, every rating explained by tactics (the consolidated capability table)
4. run the **three-stage, non-compensatory match** and defend the result	Part IV: knock-out screening, veto rule, holistic ordinal reading; hybrids are the normal case
5. record decisions as **ADRs** and attach a **measurement contract**	Part IV: the procedure ends in an ADR, every ADR in a measurement contract – A5, a decision is a hypothesis tested over the life cycle
6. engineer **AI components** into a system (Axis B) and use **AI tools** in the process (Axis A) with the same discipline	Part V: A6 held – one class, one dimension, five cell shifts, one composition pattern; Axis A raised the value of the artefacts

Open book: the map and the four tables

Your map: Maxim 8's one sentence



Part V: AI changes no step of it

The four tables the argument runs on

- the **twelve dimensions** (Part I)
- the **consolidated capability table** (Part II)
- the **requirements table** of the ten classes (Part III)
- the 7×10 **fit matrix** (Part IV)

Part V's additions

- the **D12 row with rationales**; the **agent capability-profile sketch**; the **task-signature table**; the **OWASP table**

The spine: the six load-bearing assumptions **A1–A6** – and the Maxims, up to Maxim 8

Focus (Lecture 1): **architecture reasoning – profiles, matching, trade-offs**; in the words of the Section 43 key concept: *name, for any cell movement, the quality-attribute mechanism behind it.*

Q&A – your questions, and one last discussion

Discussion

Your questions first – on the script, the four tables, the procedure, the examination. Then close the module with the symmetry of the previous section, and with the limits of the theory – the artefact pairs across the two axes:

Axis A artefact	Axis B artefact
`AGENTS.md`	the system prompt and ontology
the CI fitness function	the runtime ontology guard
the code-review obligation	the human-oversight duty of the AI Act
the test suite as the agent's operating licence	the eval harness as the model's operating licence

- For each pair: is this genuinely *one* engineering discipline observed at two binding sites – or are there differences of kind, in **reversibility**, in **accountability**, in **failure semantics**, that the symmetry hides?
- And if it is one discipline: which parts of this script would survive the next order-of-magnitude improvement in model capability – and which would you expect to rewrite?

The format: three lessons, one milestone

Week 14 in the semester plan: **1 lesson synthesis + 3 lessons presentations, architecture defence, and peer reviews** – deliverable **A3: final presentation**, milestone **M6**.

Lesson	Block	Content
1	Synthesis and exam hints	recap of the synthesis and exam orientation; Q&A – *no new material*
2–4	Presentations, defence, peer reviews	each team: **present the system**, **defend the architectural trade-offs**, and **reflect** on where AI helped and where it hurt; presentations and defences are **peer-reviewed**

Slot length per team, presentation order, and the peer-review form: to be announced.

Project Link: Portfolio Intelligence Platform

M6 – Presentation and Architecture Defence (Week 14). Present the system and **defend your architectural trade-offs**. Reflect on where AI helped and where it hurt – **in building (A) and in the system (B)**.

What to present: the system the mandatory part requires

Present the system (M6) – the one the mandatory part of the exercise sheet requires:

1. Architecture **derived from the ontology**, documented with **ADRs** and a **C4-style diagram**
2. **Deterministic services** with full, exact tests against the reference vectors
3. **Orchestrated advisor** with 2–3 specialised sub-agents cooperating only via service contracts
4. **Ontology guard** on all AI insights; **resilience** against external-data outages
5. **Evaluation harness** for the AI component; basic **observability** (cost and latency)

Important Note

- Use a **snapshot** of market/news data so your demo and grading are reproducible even if the live APIs misbehave
- Keep the deterministic core **free of LLM calls** – *this is the line that is graded*
- Distinction work (autonomous planning, self-repair loops, model routing, CI/CD, tracing): every topology escalation justified against the task-signature table and priced per the default rule (Lecture 13)

The architecture defence: defend your trade-offs

The trade-offs you defend are the ones recorded in **A2** – the **ADR**, the **C4 diagram**, the **measurement contract**. The project assessment emphasises:

Emphasis	What is assessed
Architecture & trade-offs	quality of decomposition, contracts, deterministic/non-deterministic separation, ADRs
Robustness	resilience, guards, graceful degradation
Quality	tests for deterministic services, eval harness for AI
AI integration	correct anti-corruption layering and ontology guarding
Operation	observability of cost/latency

Key Concept

A *defence* in the sense of learning objective 4: **run the three-stage, non-compensatory match and defend the result**; the decision as an **ADR**; every ADR ends in a **measurement contract** – a decision is a hypothesis tested over the life cycle (A5).

Reflection: where AI helped and where it hurt – Axis A and Axis B

The second half of M6 is the reflection – on **both axes** of Part V, in the vocabulary of the synthesis:

Axis A – AI in building the system

- where agentic tools helped – and where the **verification bottleneck** bit (Axis A: *generation* is cheap, *verification* is the bottleneck)
- which of your artefacts served as agent context: **documentation as agent context**, **fitness functions as operating licence**

Axis B – AI in the system

- where the LLM component helped – and where the **validation bottleneck** bit (Axis B: *model output* is cheap, *validation* is the bottleneck)
- what contained it: **contracts**, **ontology guards**, **eval harnesses**, and a **human-owned determinism boundary**

Key Concept

The reflection has a thesis to test against your own project: **one discipline at two binding sites** – containing cheap, fallible generation behind expensive, explicit verification. Did it hold in *your* build and in *your* system? Where did it not?

Peer reviews: the third element of lessons 2–4

- Presentations and defences are **peer-reviewed** – the semester plan's week 14: *presentations* + *architecture defence* + *peer reviews*, the third element of lessons 2–4
- **Peer-review form, criteria, and reviewer assignment: to be announced**
- Every team has solved the *same* mandatory part with the *same* procedure – differences between the systems are differences of *decision*, and a decision is what a defence exposes

Discussion

A listening prompt, not a review form. Listening to another team's defence: which High weight of $R(a)$ carried their decision, and which veto did they have to mitigate? Would *your* measurement contract have caught the risk they name?

Assessment of the project (50 % of the module grade)

The project counts **50 %** of the module grade – including:

- the **requirements dossier** (deliverable A1, week 3)
- the **architecture dossier** with ADR and measurement contract (deliverable A2, week 7)
- the **implementation** (weeks 8–13, milestones weeks 9/11/13)
- the **final presentation with architecture defence** (deliverable A3, week 14)

Evaluation emphasises

- **Architecture & trade-offs**
- **Robustness**
- **Quality**
- **AI integration**
- **Operation**

Distinction (Kür) – for top grades

- autonomous planning
- self-repair loops
- model routing with cost/latency reported
- CI/CD and richer observability (tracing)

The remaining 50 % is the written module examination (60 minutes, open book, closed internet).

The last milestone: deliverable A3 – then the written examination

Deliverable A3 checklist (M6)

- the system, presented along the mandatory part – reproducible demo on a data snapshot
- the architecture defence on A2's artefacts: ADR, C4, measurement contract
- the reflection: where AI helped and where it hurt – in building (A) and in the system (B)

and: peer reviews of the other presentations (lessons 2–4)

The written examination

- 50 %; end of semester; **60 minutes**
- **open book**: script and own notes, on paper; **closed internet**
- focus: **architecture reasoning – profiles, matching, trade-offs**

Reading

- **the whole script, Parts I–V** – the four tables and Maxim 8's one sentence as the map



Thank you!

Dr. Florian Herzog

Fachhochschule Graubünden, Chur

AISE502 – AI in Software Engineering II