



id: SI_004

model: openai/gpt-5.4-mini

seeds:

04_risiko: mulighet

03_relasjon: tillit

05_handling: vent

01_perspektiv: operasjonell

02_tidhorisont: langsiktig

Instans SI_004

Seeds: operasjonell | langsiktig | tillit | mulighet | vent

Den største blindsonen er å behandle studieplanen som et **innholdsproblem** når den egentlig er et **kapasitets- og tillitsproblem**. Med bare 10–12 emner over tre år må programmet velge hva studentene skal kunne **gjøre trygt og selvstendig** etter endt løp, ikke bare hva de har “vært innom”.

Tre blinde soner

1. For mye bredde, for lite operasjonell mestring

- Matematikk, fysikk og systemforståelse er viktige, men hvis de ikke kobles til praktiske utviklingsløp, blir de en buffer som fortrenger ferdigheter arbeidsgiverne ser umiddelbart.
- Blindsonen er å undervurdere at mange studenter allerede møter industrien med forventning om å levere i **Git, sky, testing, API-er, sikkerhet og samarbeid**.

2. For mye spesialisering, for lite varighet

- AI-verktøy og populære rammeverk endrer seg fort. Å fylle planen med verktøyspesifikk opplæring kan gi kortsiktig attraktivitet, men lav langsiktig verdi.
- Det riktige er å prioritere **prinsipper**: programvarearkitektur, dataflyt, evalueringsforståelse, modellbegrensninger, versjonskontroll og kvalitetssikring.



3. Autonomi blir ofte antatt, ikke trent

- Studenters evne til å lære autonomt utvikles ikke automatisk. Den må bygges med strukturerte “vent”-soner: oppgaver der de må tåle uklarhet, feile, søke kunnskap og dokumentere valg.
- Blindsonen er å tro at “de lærer dette underveis” uten eksplisitt design.

Hva bør veies mot hverandre?

- **Bredde vs. dybde:** Jeg ville heller valgt **smalere bredde med tydelig kjerne** enn mange emner som hver blir litt tynne.
- **Verktøy vs. varighet:** Prioriter 2–3 stabile kompetanseområder over mange flyktige verktøy.
- **Lærerstyring vs. studentautonomi:** For lite støtte gir frafall; for mye støtte gir avhengighet. Målet er gradvis overgang fra tett veiledning til selvstyrt arbeid.

Konsekvens for programdesign

Et godt kompromiss kan være:

- 3 emner grunnlag: matematikk/systemforståelse
- 4–5 emner software engineering-kjerne
- 1–2 emner AI/data med fokus på anvendelse og vurdering
- 1–2 emner prosjekt/praksis med høy autonomi og industrinær problemstilling

Neste steg

1. Gjennomfør en **kompetansekartlegging** mot 10–15 konkrete roller i regionen.
2. Identifiser **hva som kan fjernes** før man legger til nytt: hvert emne må forsvare sin langsiktige verdi.
3. Design ett emne med eksplisitt mål om **autonom læring**: problemdefinisjon, kunnskapssøk, refleksjon og teknisk beslutningslogg.