



id: SI\_006

model: openai/gpt-5.4-mini

seeds:

04\_risiko: sårbarhet

03\_relasjon: konkurranse

05\_handling: vent

01\_perspektiv: operasjonell

02\_tidhorisont: historisk

## Instans SI\_006

---

**Seeds:** operasjonell | historisk | konkurranse | sårbarhet | vent

Det operasjonelle spørsmålet er ikke bare hva studentene skal lære, men hva de faktisk rekker å bli gode på med 10–12 emner. Den historiske fellen er å bygge et “mini-sivilingeniørprogram” med for mye matte/fysikk og for lite moderne utviklingspraksis; den konkurransemessige fellen er motsatt: å bli et kurs i dagens verktøy som taper mot kortvarige bootcamps og industrikurs. Blindsonen ligger ofte i **vent-faktoren**: ledelsen antar at studentene “tar igjen” manglende grunnmur eller spesialisering senere. Det gjør de sjelden systematisk.

Tre spenninger bør veies eksplisitt:

### 1. Bredde vs. relevans

Grunnleggende matematikk, fysikk og systemforståelse gir robusthet, men i et lokalt økosystem med forsvarsindustri og høyteknologi er arbeidsmarkedets signal klart: programvare, arkitektur, sikkerhet og integrasjon betyr mest. For mye fysikk kan bli historisk arv mer enn operasjonell verdi. Spør: hvilke 2–3 emner gir reell ingeniørbredde uten å spise halve graden?

### 2. Spesialisering vs. varighet

AI-verktøy og software engineering er etterspurt, men verktøy blir raskt foreldet. Prioriter derfor prinsipper og praksis: testing, versjonskontroll, CI/CD, krav, arkitektur, datastrukturer, modellforståelse, MLOps-grunnlag. Et emne i “AI-verktøy” uten evaluering av begrensninger blir sårbart og kortlivet.

### 3. Kompetanse vs. autonom læring

Den største blindsonen er å anta at selvstendig læring kommer av seg selv. Autonomi må trenes operasjonelt: prosjektarbeid med ukjente rammer, krav om å lese dokumentasjon, skrive tekniske notater, og refleksjon over metodevalg. Dette er konkurransefortrinnet som ikke kopieres lett.



Mitt avvikende råd: ikke la “AI” bli et eget tyngdepunkt før basisen er sterk. Bygg heller et **software-first-program** med innfelt AI, sikkerhet og systemtenkning. For et 10-12-emners løp kan en grov fordeling være: 3 kjerneemner i programmering/systemutvikling, 2 i matematikk/diskret, 1 i fysikk/ingeniørforståelse, 1 i sikkerhet, 1 i databaser/nettverk, 1 i arkitektur/DevOps, 1 i AI/data, og 1-2 prosjekt/emneintegrasjoner.

**Neste steg:**

1. Kartlegg 5-7 konkrete stillingsannonser fra regionen og oversett dem til læringsutbytter.
2. Fjern emner som ikke støtter minst to av tre: bredde, relevans, autonomi.
3. Innfør et obligatorisk capstone-prosjekt med eksterne krav og ukjent verktøysett.