



id: SI_003

model: openai/gpt-5.4-mini

seeds:

04_risiko: sårbarhet

03_relasjon: konkurranse

05_handling: utforsk

01_perspektiv: strategisk

02_tidhorisont: langsiktig

Instans SI_003

Seeds: strategisk | langsiktig | konkurranse | sårbarhet | utforsk

Det strategiske spørsmålet er ikke bare hva studentene skal lære, men hva slags kompetansekjede programmet skal produsere for et miljø der forsvar, industri og høyteknologi krever både leveranseevne og tillit. Med bare 10-12 emner over 3 år er den største risikoen en **for tynn og fragmentert plan**: litt matematikk, litt AI, litt programmering – men ingen tydelig bærekraftig kjerne.

Sentrale blindsoner

1. **Overvektning av verktøy, undervurdering av grunnmur** AI-verktøy og moderne stacks endrer seg fort. Hvis studieplanen låser seg til konkrete rammeverk, kan den være utdatert før studenten uteksamineres. Den langsiktige verdien ligger i algoritmisk tenkning, programvarearkitektur, datastrukturer, modellering og testbarhet.
2. **For lite system- og sikkerhetsforståelse** Nærhet til forsvarsindustri gjør sårbarhet til et konkurransepunkt. Mange programmer undervurderer secure coding, trusselmodellering, tilgangsstyring, sporbarhet og systemintegrasjon. Dette er ikke “tillegg”; det er differensierende kompetanse.
3. **Autonomi som antakelse, ikke læringsmål** Studentenes evne til å lære selvstendig må trenes eksplisitt. Uten opplæring i litteratursøk, eksperimentdesign, dokumentasjon, kodegjennomgang og refleksjon blir de avhengige av det som er kjent i dag, ikke i morgen.



Prioriteringsargumenter

- **For bredde:** Matematikk/fysikk gir robusthet, men kan bli en flaskehals hvis det spiser for mye av 10–12 emner. Bredde må være målrettet, ikke generell.
- **For spesialisering:** Software engineering og AI gir attraktivitet og arbeidsmarkedsrelevans, men bør bygges på en kjerne av engineering-prinsipper. Ellers blir kandidaten “verktøydrevet” og sårbar.
- **For autonomi:** Dette er den mest undervurderte konkurransefordelen. Et program som trener kontinuerlig læring, gjør kandidater mer verdifulle over en 10–20 års karriere.

En mulig balanse

Tenk 4 emner kjerneprogrammering/systemer, 2 emner matematikk/modellering, 2 emner software engineering/CI/CD/testing, 1 emne AI/ML, 1 emne sikkerhet/etikk, 1–2 prosjekt- eller integrasjonsemner.

Neste steg

1. Kartlegg hvilke læringsutbytter som faktisk kan kuttes uten å svekke kandidatens langsiktige verdi.
2. Lag en matrise som rangerer emner etter: relevans om 5 år, relevans for regional industri, og sårbarhet for teknologisk foreldelse.
3. Innfør ett obligatorisk “autonomi-emne” med søk, lesing, eksperimentering og faglig selvregulering.