



id: SI\_005

model: openai/gpt-5.4-mini

seeds:

04\_risiko: usikkerhet

03\_relasjon: konkurranse

05\_handling: implementer

01\_perspektiv: strategisk

02\_tidhorisont: kortsiktig

## Instans SI\_005

**Seeds:** strategisk | kortsiktig | konkurranse | usikkerhet | implementer

Den største blinden er å tro at dette er et rent **faglig** designproblem. Det er i realiteten et **strategisk konkurransevalg** under usikkerhet: Programmet konkurrerer ikke bare mot andre bachelorløp, men mot bachelorer som lover "AI", industri som forventer umiddelbar produktivitet, og en arbeidsmarkedssituasjon som kan skifte raskere enn en 3-årig plan.

### Tre blindsoner

- Overvektig av "moderne" verktøy** AI-verktøy og rammeverk kan gi kortsiktig attraktivitet, men de har kort halveringstid. Hvis 3-4 emner blir tool-spesifikke, risikerer dere å utdanne studenter som er sterke i dagens stack, men svake i neste revisjonssyklus. Det er en klassisk implementeringsfelle: man optimaliserer for synlighet nå, ikke robusthet senere.
- Undervurdering av grunnmuren** I et miljø med forsvarsindustri og høyteknologi er det lett å anta at "software engineering" er nok. Men systemforståelse, diskret matematikk, numeriske metoder, signal-/sensorforståelse og grunnleggende fysikk er konkurransefortrinn fordi de gjør kandidater anvendelige i tverrfaglige, sikkerhetskritiske og systemintensive roller.
- Autonomi som pynt, ikke struktur** Mange program sier de vil lære studenter å lære selv, men gir ikke eksplisitt trening i det. Uten krevende, åpne oppgaver, refleksjon over læringsstrategier og krav om dokumentert selvstendig oppdatering, blir "autonom læring" et slagord.



## Argumenter som bør veies mot hverandre

- **Bredde først** gir robusthet, men kan oppleves som mindre attraktivt på kort sikt.
- **Spesialisering i software/AI** gir rekrutteringskraft og næringstilknytning, men øker risikoen for foreldelse.
- **Autonomi som kjernekompetanse** reduserer usikkerhet over tid, men krever plass i emnene og kan være vanskelig å måle.

## Et mulig prioriteringsgrep

Med bare **10-12 emner** bør programmet ha:

- **4-5 emner grunnmur:** matematikk, programmering, datastrukturer/algoritmer, systemforståelse, fysikk/numerikk
- **3-4 emner software engineering og AI**
- **2 emner som trener autonomi:** prosjektbasert arbeid, faglig fordypning, teknologiendrings-/verktøy-evaluering
- resten som **integreerte industrinære prosjekter**, ikke flere løse spesialiseringsemner

## Neste steg

1. Kartlegg hvilke 3-4 læringsutbytter som må overleve neste revisjonssyklus uansett teknologi.
2. Bygg en emnematrise som viser hvilke emner som dekker grunnmur, spesialisering og autonomi — og hvor det er overlapp.
3. Test planen mot 2 scenarier: "AI-boom" og "AI-tilbakeslag", og se om kandidaten fortsatt er attraktiv i begge.