



id: SI\_001

model: openai/gpt-5.4-mini

seeds:

04\_risiko: sårbarhet

03\_relasjon: makt

05\_handling: implementer

01\_perspektiv: strategisk

02\_tidhorisont: syklisk

## Instans SI\_001

---

**Seeds:** strategisk | syklisk | makt | sårbarhet | implementer

Studieledelsen bør lese dette som et **porteføljeproblem under knapphet**, ikke som et pedagogisk idealvalg. Med bare **10-12 emner** over tre år kan man ikke maksimalisere både bredde, dybdespesialisering og “framtidssikkerhet” samtidig. Derfor må planen designes for **syklisk endring**: noen emner må være stabile, andre må være lett utskiftbare.

---

### Blindsoner å passe seg for

1. **Overvekting av dagsaktuell teknologi** AI-verktøy og spesifikke rammeverk kan gi kortsiktig attraktivitet, men blir raskt utdaterte. Blindsonen er å tro at “relevant nå” er det samme som “robust om 3-5 år”. Et bachelorprogram bør ikke bygge identitet rundt verktøy som kan være borte før neste revisjonssyklus.
2. **For svak systemforståelse** I en industriell kontekst med forsvar og høyteknologi er det farlig å utdanne “kodeoperatører” uten forståelse for krav, arkitektur, sikkerhet, test og drift. Maktforholdet her er tydelig: arbeidsgiverne vil ha folk som kan levere, men samfunnet trenger kandidater som kan vurdere konsekvenser.
3. **Autonomi undervurderes som kjernekompetanse** Mange programplaner antar at selvstudie skjer av seg selv. Det er en sårbarhet. Hvis studentene ikke trenes i å lære nye språk, verktøy og domener alene, blir de avhengige av at utdanningen alltid “treffer riktig” teknologi.



---

## Argumenter som bør veies mot hverandre

- **Breddeargumentet:** Matematikk, fysikk og ingeniørforståelse gir varig analytisk kapasitet og gjør kandidaten mobil mellom roller.
- **Spesialiseringsargumentet:** Software engineering, sky, DevOps, AI-assisterte arbeidsformer og sikker utvikling gir direkte relevans mot regionalt arbeidsmarked.
- **Autonomiargumentet:** Et program bør optimaliseres for læringshastighet, ikke bare kunnskapsmengde. Kandidaten må kunne retool'e selv.

---

## Strategisk anbefaling

Velg en **stabil kjerne + roterende topp**:

- **Kjerne (6-7 emner):** programmering, algoritmer/datastrukturer, matematikk, systemutvikling, databaser, operativsystem/nettverk, software engineering.
- **Roterende/valgfrie emner (3-4 emner):** AI-verktøy, sikkerhet, sky/DevOps, embedded/industrielle systemer, UX eller dataanalyse.

---

## Neste steg

1. Kartlegg hvilke emner som faktisk gir **varig kompetanse** versus **verktøyspesifikk trening**.
2. Lag en **kompetansematrise**: hver emneplass må forsvare enten bredde, spesialisering eller autonomi.
3. Etabler en **2-årig revisjonsmekanisme** for valgfrie/små emner, mens kjernen holdes stabil i minst 4-5 år.