



id: SI\_003

model: deepseek/deepseek-chat-v3

seeds:

04\_risiko: sårbarhet

03\_relasjon: tillit

05\_handling: utforsk

01\_perspektiv: operasjonell

02\_tidhorisont: kortsiktig

## Instans SI\_003

**Seeds:** operasjonell | kortsiktig | tillit | sårbarhet | utforsk

### Analyse:

Fra et **operasjonelt** og **kortsiktig** perspektiv handler balansen mellom kvalitet og effektivitet om å ta grep som gir umiddelbar effekt uten å ofre tillit eller skape langsiktig sårbarhet. Mange fokuserer på “perfeksjon” eller “raske løsninger,” men nøkkelen er å **utforske** grensen der små justeringer gir store gevinster.

Eksempel: En utvikler som bruker 80 % av tiden på å finpusse en funksjon som kun 10 % av brukerne vil merke, taper tid på bekostning av andre oppgaver. I stedet kan en **kortsiktig** MVP-test (Minimum Viable Product) med klar kommunikasjon om at dette er en **utforskende** fase, bygge **tillit** gjennom åpenhet, samtidig som man unngår **sårbarhet** ved å investere for mye i én løsning.

Tall: 70/30-regelen. Bruk 70 % av tiden på “god nok” kvalitet, 30 % på å evaluere og justere. Dette gir operasjonell fleksibilitet.

### Dissens:

“Bare gjør det bra fra starten” er ofte ineffektivt. **Sårbarhet** (å innrømme at noe kan være uferdig) og **tillit** (at teamet kan iterere) skalerer bedre enn å late som alt må være perfekt.

### Neste steg:

1. **Introduiser “tidshaker”:** Sett en fast tid (f.eks. 25 minutter) per oppgave for å unngå overoptimalisering.



2. **Åpne statusrapporter:** Del ukentlig hva som er “god nok” og hva som krever mer tid, for å styrke **tillit**.
3. **Test én “risikabel” effektivisering:** Velg én oppgave der du bevisst bruker 30 % mindre tid, og mål resultatet.

SI\_003 avslutter her.