



Produktorientering og autonome team

Berit Bakken (NAV)

Ellen Daleng (Computas)

NOKIOS 29. oktober 2019

13:00 - 17:00

HØRA

Hensikt

Utforske hva som kreves av en organisasjon som ønsker å legge til rette for produktutvikling i autonome team

Agenda

- Innledning og oppvarming
- Organisering og avhengigheter
- 15.00 - 15.30: Pause
- Hva kreves av omgivelsene?
- Oppsummering og avslutning

Ønsket resultat

- Innsikt i fordeler og ulemper ved ulike måter å dele opp virksomheten og teamene på
- Innsikt i hva som kreves av omgivelsene for å gi teamene optimale forhold

Kjøreregler

- Når vi som fasilitatorer ønsker oppmerksomhet, rekker vi hånda i været
- I gruppa tar dere ansvar for at:
 - alle kommer til orde
 - diskusjonene ikke sporer av
 - tidsplaner overholdes
 - innsikt dokumenteres når det går fram av oppgaven
- Flere? Bli enige i gruppa!



**Hva er autonome
team?**

Autonome team

En gruppe mennesker som jobber sammen *er ikke nødvendigvis et team...*

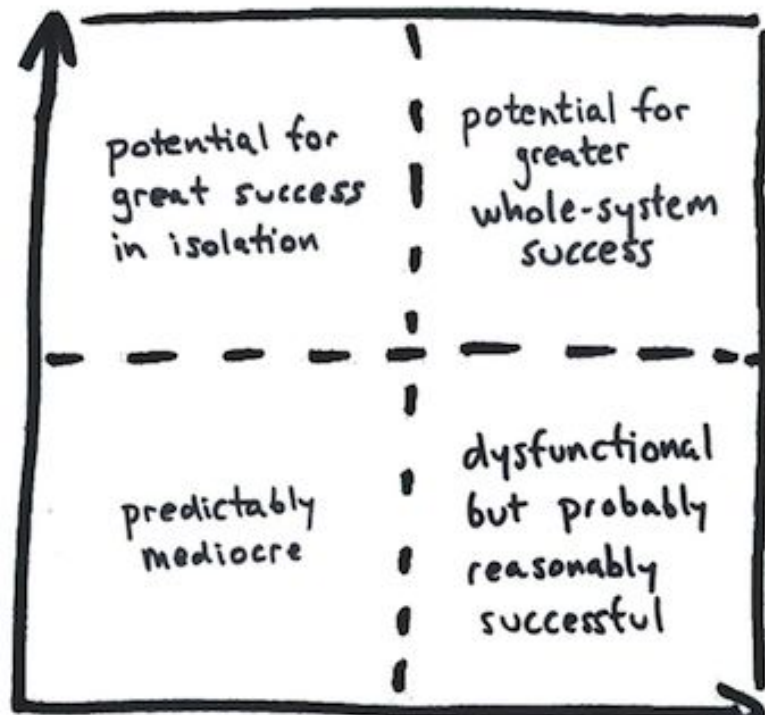


Team

en gruppe som
arbeider mot **samme mål** og er
avhengig av hverandre for å nå det

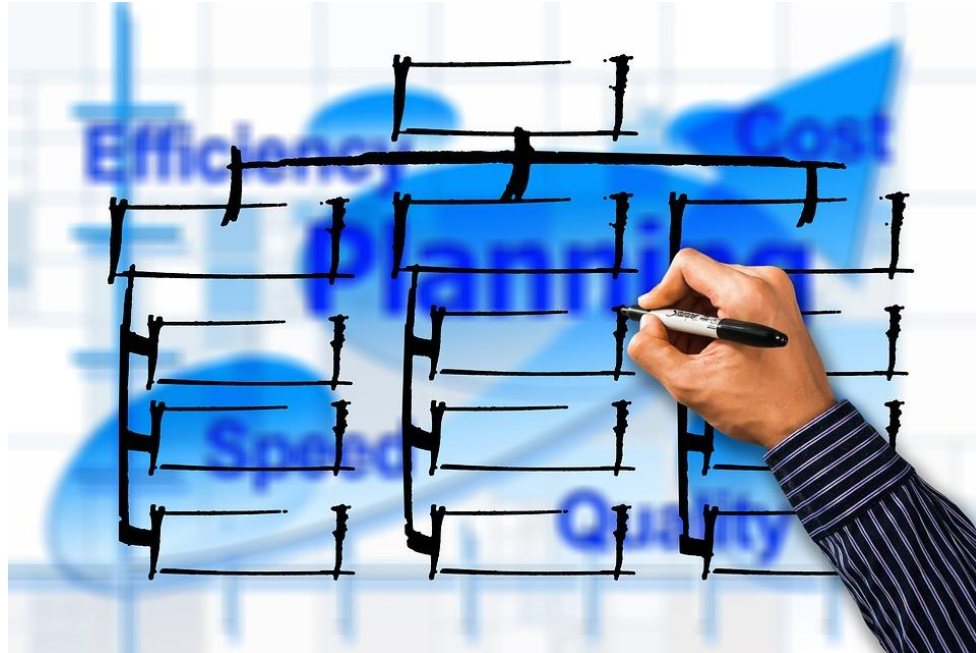
Autonomt team: Et team som styrer sine egne aktiviteter, innenfor definerte rammer

Autonomy



Team

Motivasjonen bak autonome team



*Skape balanse mellom økt kompleksitet i omgivelsene
og organisasjonens evne til håndtere den*

**Hva er produkt-
orientering?**

Prosjekt

output (leveranse)

omfang, tid, budsjett

abstrakt (arbeidsstyring)

planlegge

har en slutt

investering lover ROI

Produkt

outcome (effekt)

kontinuerlig verdi

konkret (et produkt)

tilpasse

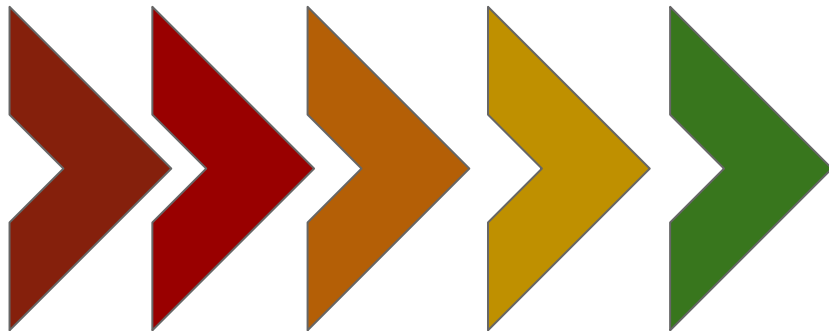
pågående

verdi driver investeringer

Organisering og avhengigheter

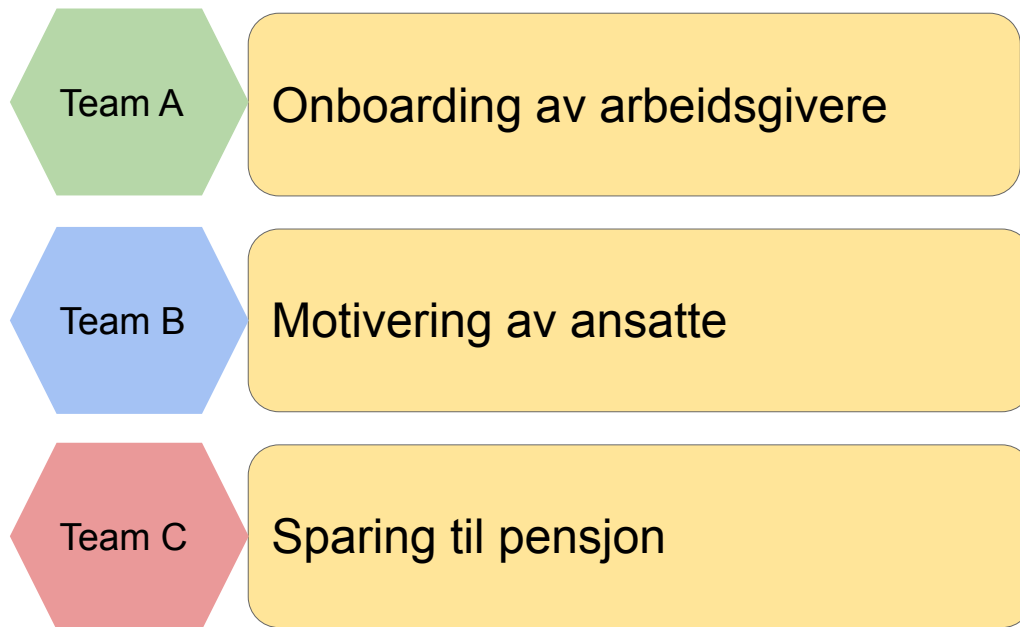
Organisering av virksomheten: Hvordan dele inn i gode “produkter?”

Avhengigheter mellom produkter
skaper avhengigheter mellom teamene:
arkitektur har stor betydning



Eksempel

Egentlig mål: **Øke utbredelsen av pensjonsprodukter**



Teamtopologi

Kognitiv belastning (Cognitive load)

Den totale mentale belastningen på arbeidsminnet

“alt du må ha kunnskap om for å utføre noe”

Tre nivåer:

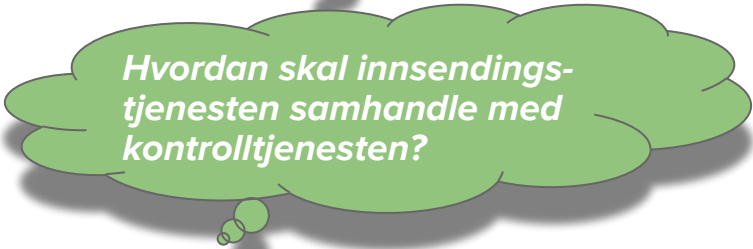
1. **Fundamental kunnskap**
du må ha
2. **Kunnskap om miljøet rundt oppgaven**
du forsøker å få gjort
3. **Oppgavekunnskap**
- det du egentlig forsøker å få til



Hvordan defineres en klasse i Java?



Hvordan var det nå jeg deployer denne komponenten igjen?

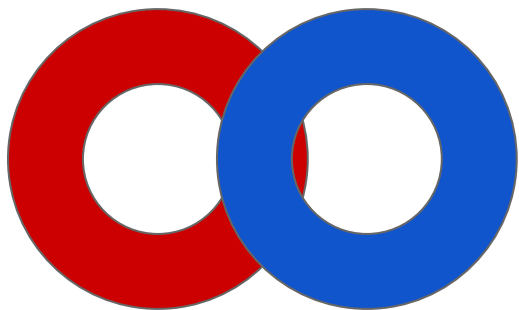


Hvordan skal innsendings-tjenesten samhandle med kontrolltjenesten?

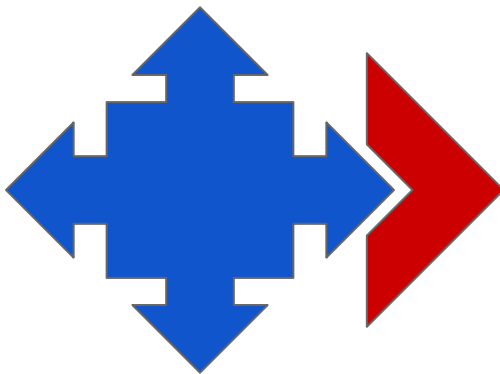
Redusere cognitive load - 3 tips

1. Skap veldefinerte interaksjonsmønstre mellom team
2. Bruk autonome *strøm-innrettede* team
3. Bygg en “tynneste nyttige plattform”
(Thinnest viable platform)

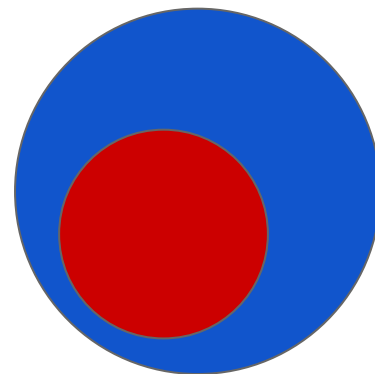
Veldefinerte interaksjonsmønstre mellom team



Samarbeid



X-as-a-service



Fasilitering

Ulike typer team

1. **Strøm-innrettet** (Stream-aligned) 
2. **Evnebygger** (Enabling) 
3. **Komplekst subsystem** (Complex subsystem) 
4. **Plattform** (Platform) 

- Fra ideer klekkes ut til produkter går på lufta og driftes videre



2 - Evnebyggende team

- Bygger de andre teamenes evne til å utføre oppgavene sine og tilrettelegger for god flyt mellom team
- Fasilitatorer, coacher, tilretteleggere



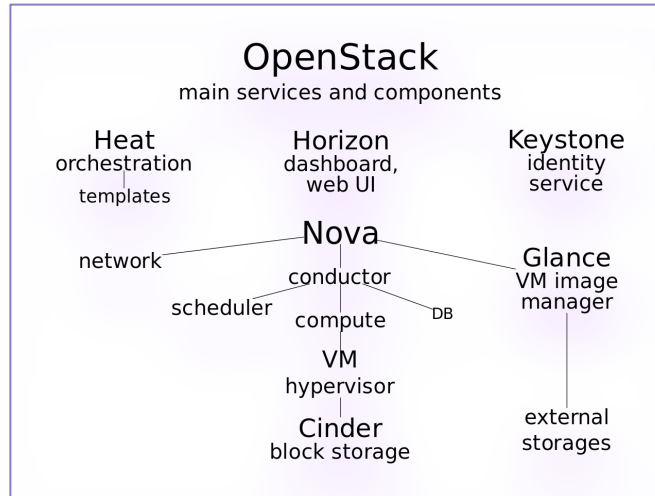
3 - Komplekst subsystem-team

- Håndterer ett eller flere subsystemer som krever spesiell kompetanse
- Letter de andre teamenes kognitive belastning
- Trengs kun i spesielle tilfeller hvor svært detaljert ekspertise er påkrevd



4 - Plattform-team

- Har ansvaret for å vedlikeholde og videreutvikle en plattform med verktøy og metoder slik at de andre teamene kan bruke denne
- Samhandler som regel etter *X-as-a-service* mønsteret



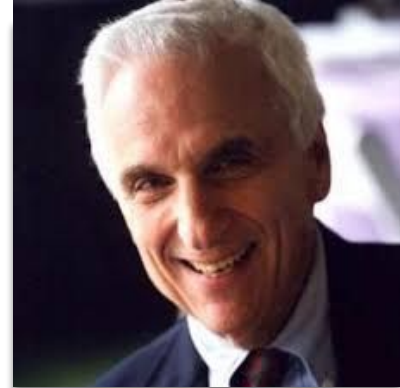
Tynneste Nyttige Plattform

Det minste settet av APIer, dokumentasjon og verktøy som er til hjelp for team som utvikler nye tjenester og systemer

- DevEx
- Enkelt i bruk
- Enkle verktøy
- Rik dokumentasjon

Bygg og forvalt plattformen som et produkt i seg selv, med strøminnrettede team som interne kunder

Conway's Lov



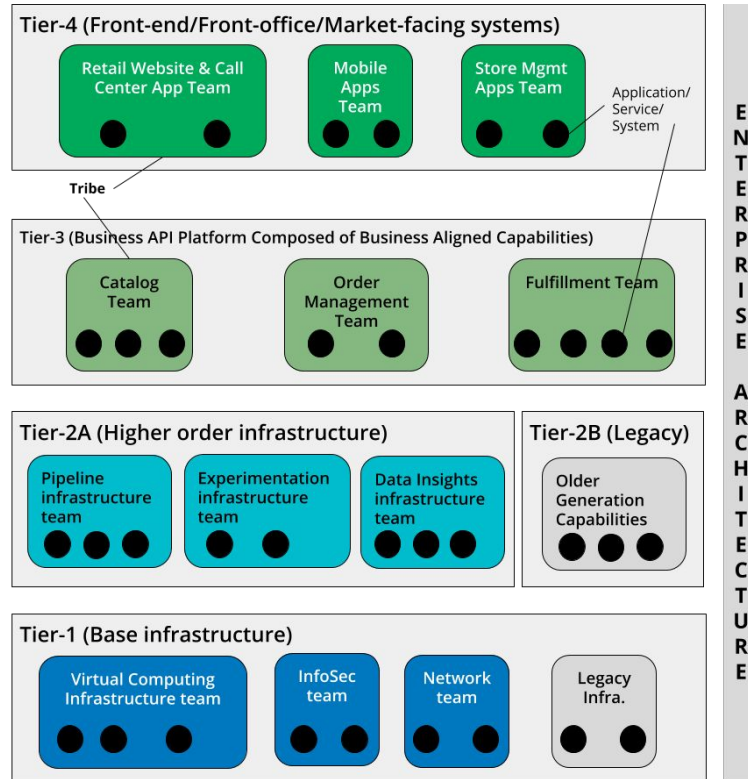
**Softwarearkitektur vil
gjenspeile kommunikasjonsmønstrene
i organisasjonen som lager den**

Bruk dette bevisst i teamorganiseringen (*Inverse Conway Maneuvre*)

Hvilken arkitektur vil du ha? Tenk “team-først arkitektur”

If the desired theoretical system architecture does not fit the organizational model, then one of the two will need to change

Team Topologies



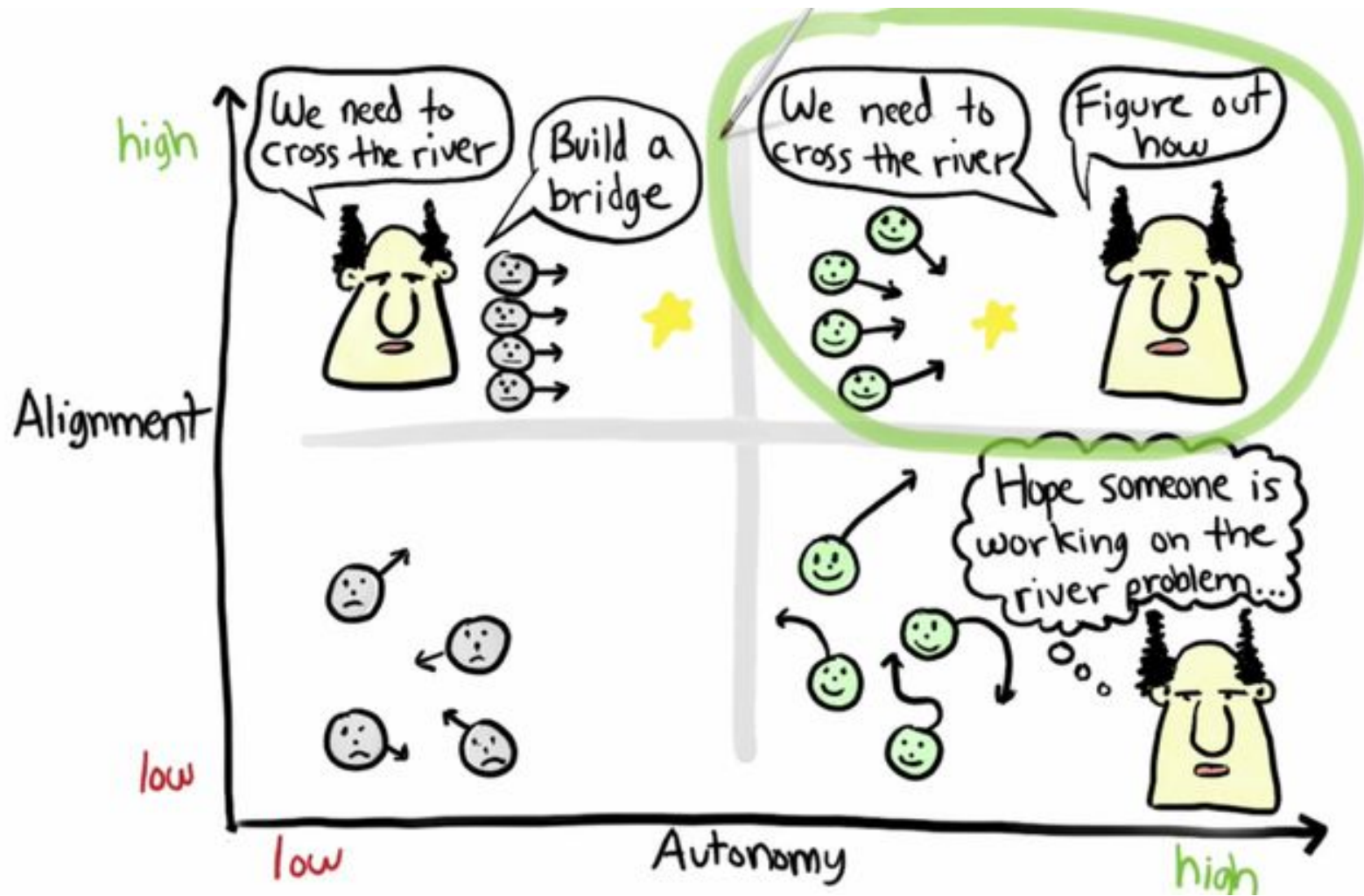
Eksempel på virksomhedsarkitektur og teaminndeling

**Hva kreves av
omgivelsene?**

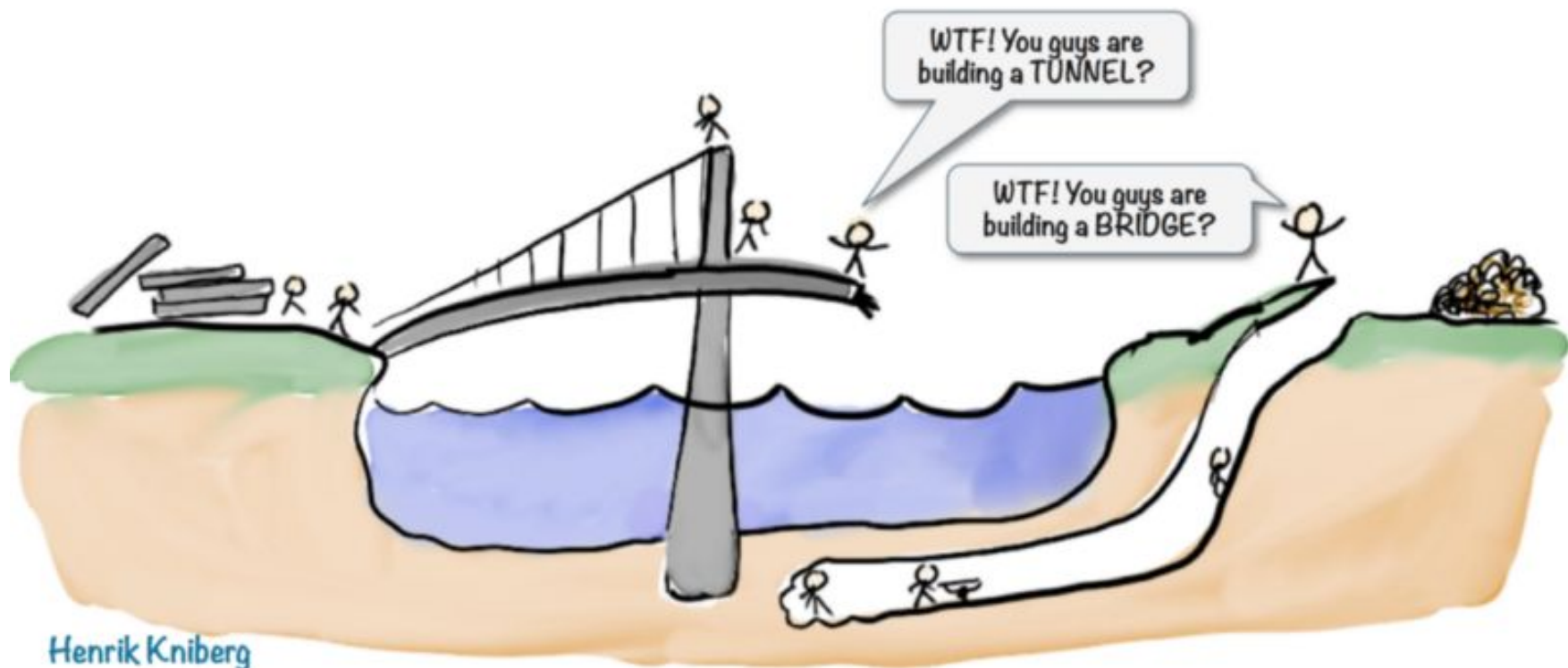
Hva karakteriserer høytpresterende team?



Autonomi og retningsstyring

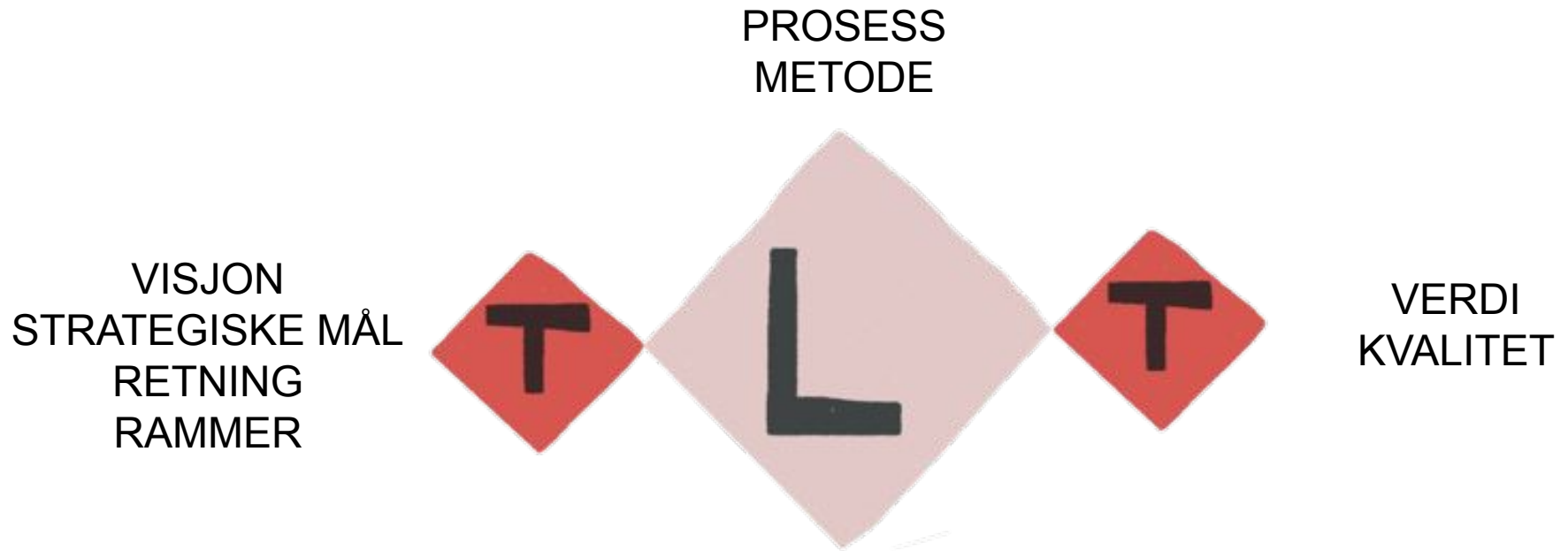


Misalignment



Henrik Kniberg

Tight loose tight



Eksempel: NAVs prinsipper for tverrfaglige produktteam



Vi setter oss inn i
brukerens situasjon



Vi realiserer
produktområdets mål



Vi eksperimenter, måler
og lærer raskt



Vi er åpne og
samarbeider med andre



Vi skaper tillit



Vi leverer verdi ofte

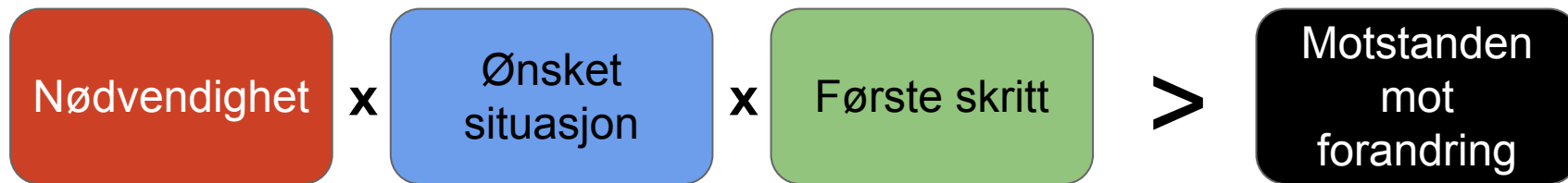


Vi er selvstyrte ...
... innenfor gitte rammer

Veien mot målet

Forandringens formel

$$N * \emptyset * F > M$$



NB! Venstre side er et produkt!

Hva kan du sette i gang i egen organisasjon?

- Hva tilsier at endring er **nødvendig**?
- Hvem skal du snakke med om hvilke mål dere **ønsker å oppnå**?
- Hva er et mulig **første skritt**?
- Hvilke **hindrende krefter** må vi svekke?
- Hvilke **drivende krefter** kan vi styrke?

Bøger og artikler

Outcomes over Output - Why Customer Behaviour is the Key Metric for Business Success (Joshua Seiden, 2019)

Team Topologies - Organizing Business and Technology Teams for Fast Flow (Mathew Skelton og Manuel Pais, 2019)

Project Myopia - Why Projects Damage Software og **Continuous Digital - An Agile Alternative to Projects for Digital Business** (Allan Kelly, 2018)

The Fractal Organization - Creating Sustainable Organizations with the Viable System Model (Patrick Hoverstadt, 2008)

Products over Projects: <https://martinfowler.com/articles/products-over-projects.html>

Business Capability centric teams: <https://martinfowler.com/bliki/BusinessCapabilityCentric.html>

Googles undersøkelse - Hva skaper høytpresterende team? <https://rework.withgoogle.com/print/guides/5721312655835136/>

Om alignment vs autonomy: https://www.youtube.com/watch?v=_qlh2sYXcQc

Spotify culture Part 1: <https://www.youtube.com/watch?v=Yvfz4HGtoPc>



**TAKK
FOR
OSS!**