

DOMENEDREVET SOSIOTEKNISK ARKITEKTUR

EINAR W. HØST



BÆREKRAFTIGE SOSIOTEKNISKE SYSTEMER

HVA ER PROGRAMVAREUTVIKLING?

HVA ER JOBBEN VÅR?

ER PROGRAMVAREUTVIKLING VANSKELIG?

JA

OFTE ER DET VANSKELIG

OFTE **BLIR** DET VANSKELIG

PROGRAMMERING ER LETT

**PROGRAMVAREUTVIKLING ER
VANSKELIG**

**HVA ER DET SOM GJØR DET
VANSKELIG?**

HVORFOR BLIR DET OFTE VANSKELIGERE OVER TID?

OPPHOPNING AV KOMPLEKSITET

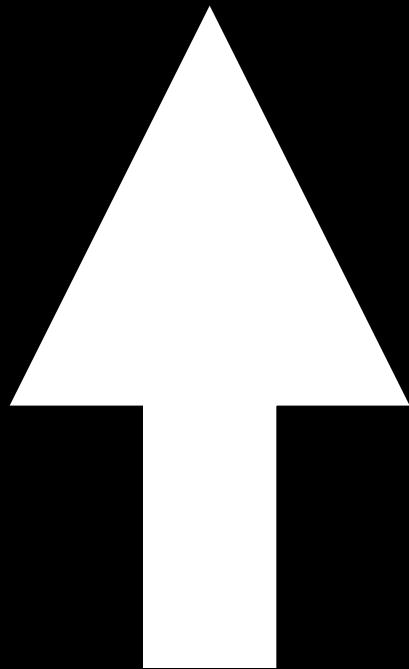
**KOMPLEKSITETEN GJØR DET
VANSKELIG Å JOBBE**

HVA FØRER TIL OPPHOPNING AV KOMPLEKSITET?

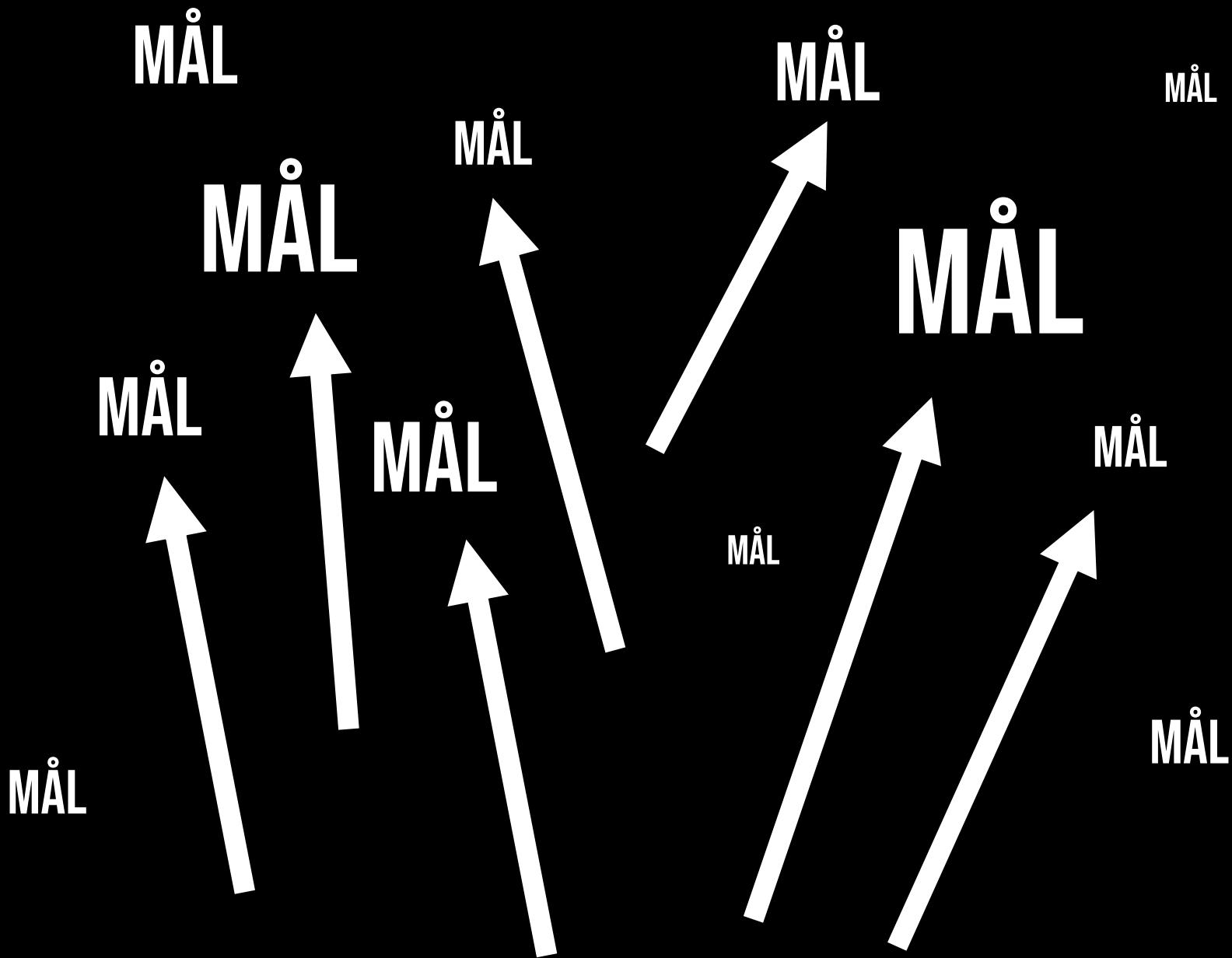
**PROGRAMVAREUTVIKLING ER
PRODUKTUTVIKLING**

HVA SKAL VI LAGE?

MÅL



START



UENIGHETER

DRAKAMPER

KJEPPHESTER

FORHANDLINGER

KOMPROMISSE

BITTERHET

OMKAMPER

**PRODUKTET KAN INNEHOLDE
SELVMOTSIGELSER**

**PRODUKTET KAN INNEHOLDE
BESLUTNINGSVEGRING**

KONTEKSTEN FOR PROGRAMVAREUTVIKLING

**PROGRAMVARE ER
SOSIOTEKNISK**

**VI BYGGER IKKE
BARE PRODUKTET**

**VI BYGGER OGSÅ
ORGANISASJONEN SOM
BYGGER PRODUKTET**

SOSIALT



TEKNISK

CONWAYS LOV

GJENSIDIG OPTIMALISERING

KONTINUERLIG UTVIKLING

KONTINUERLIG ENDRING

**PRODUKTET
ER IKKE STATISK**

**OMGIVELSEN
ER IKKE STATISK**

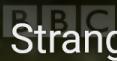
**ORGANISASJONEN
ER IKKE STATISK**

ALTER I ENDRING

EMBRACE CHANGE!

LETT FOR DEG Å SI!

ENDRINGENE OMFAVNER OSS



Strangled by a Boa Constrictor | Deadly 60 | Series 2 | BBC Earth



until eventually I can't get
any air in or out of my lungs

SUBSCRIBE

▶ ▶| 🔍 0:58 / 2:33

▼

▶ CC HD #

ENDRING ER KONSTANT

HVORDAN KAN VI HÅNDTERE ENDRING BEDRE?

KAN DDD HJELPE?

Domain-Driven DESIGN

Tackling Complexity in the Heart of Software



Eric Evans
Foreword by Martin Fowler

MEN HVORDAN?

**DOMENET ER
PROGRAMVARENS HJERTE**

HOLD DEG NÆR DOMENET

DDD is a design discipline where you

- **Grasp the domain**
- **Agree on a language**
- **Express it in shared models**
- **Embrace complexity**
- **Separate models in contexts**
- **... and evolve them continuously**

NOEN ADVARSLER

DDD ER IKKE EN MIRAKELKUR

JOBBEN ER JOBBEN

**ALDRI GJØR DDD
FOR Å GJØRE DDD**

**DDD ER IKKE
[KUN] FOR UTVIKLERE**

**IKKE GJØR DDD MED MINDRE
HELE TEAMET ER MED**

HVA TILBYR DDD?



HVORFOR EVENTSTORMING?

ETABLERE KOMMUNIKASJON

MAKSIMERE LÆRING

LÆRING ER FLASKEHALSEN

**KOMMUNIKASJON ER
NØKKELEN TIL LÆRING**

HVA ER EVENTSTORMING?

EN TEKNIKK FOR BEVISST LÆRING I FELLESSKAP

#1 ALLE I ETT ROM

#2 UENDELIG PLASS TIL MODELLERING

#3 POST-ITS MED HENDELSER I DOMENET UTTRYKT I FORTID

#1 ALLE I ETT ROM

**INGEN SKJULTE
INTERESSEHAVERE**

ALLE PERSPEKTIVER PÅ BORDET

**HVIS VI SKAL LAGE NOE
SAMMEN BØR VI
KJENNE HVERANDRE**

KNYTTE ANSIKTER TIL NAVN OG ROLLER

KNYTTE FORBINDELSE

**STARTE EN SAMTALE
OM PRODUKTET**

ER DET MULIG I PRAKSIS?

**ER DET IKKE
FRYKTELIG DYRT?**

HVOR MYE KOSTER

FRAGMENTERT OG ASYNKRON

KOMMUNIKASJON

HVER ENSTE DAG?

#2 UENDELIG PLASS TIL MODELLERING

VI VET IKKE HVOR STORT
PROBLEMET ER FØR VI HAR
UTFORSKET DET

**TRANG PLASS GJØR AT
TING BLIR UTELATT**

VI VIL HA MED ALT

MINIMERE MOTSTAND

**ALLE MÅ HA
LAPPER OG TUSJER**

HVORFOR HENDELSER?

LAV TERSKEL FOR Å BIDRA

**ALLE KAN BESKRIVE EN
HENDELSE I FORTID**

**FOKUSERE PÅ DOMENET
IKKE PÅ LØSNINGEN**

HVORFOR POST-ITS?

LAV TERSKEL FOR Å BIDRA

BONUS

MAKSIMERE LÆRING

**PROGRAMVAREUTVIKLING ER
SOSIOTEKNISK**

LÆRING OM ORGANISASJONEN

LÆRING OM FOLKENE

**STUDERE
KROPPSSPRÅK**

FÅ FREM
MOTSETNINGER

**IKKE FEIE TING
UNDER TEPPET**

FÅ ALT PÅ BORDET

**INGEN MAGISK LØSNING
PÅ PROBLEMENE**

SYNLIGGJØRING AV PROBLEMENE

DREIE DISKUSJONEN MOT DOMENET

EVENTSTORMING HANDLER OM DET STORE BILDET

ALLE SKAL SNAKKE SAMMEN

ÉN STOR MODELL?

ETT FELLES SPRÅK?

BDUF?

NEI

**ALLE KAN IKKE SNAKKE
SAMMEN HELE TIDEN**

CONQUER → **EVENTSTORMING**
AND DIVIDE → **MODELLERING**
AND CONQUER → **PROFIT**

MODULARISERING

**VI MÅ PARTISJONERE
DOMENET**

VI MÅ PARTISJONERE ORGANISASJONEN

SOSIOTEKNISK ARKITEKTUR

BOUNDED CONTEXT

GYLDIGHETSMRÅDE FOR EN MODELL OG ET SPRÅK

STRATEGISK VALGT GRENSE

**UNNGÅ Å MODELLERE
HELE VERDEN**

**HIGH COHESION
LOW COUPLING**

Structured Design

Fundamentals of a Discipline of Computer
Program and Systems Design

Edward Yourdon / Larry L. Constantine



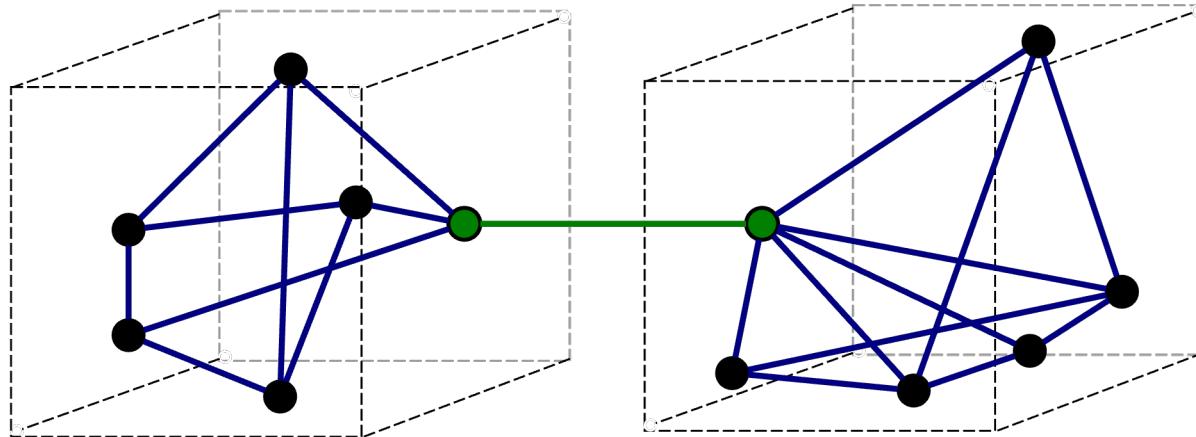
Reliable Software THROUGH Composite Design

GLENFORD J. MYERS

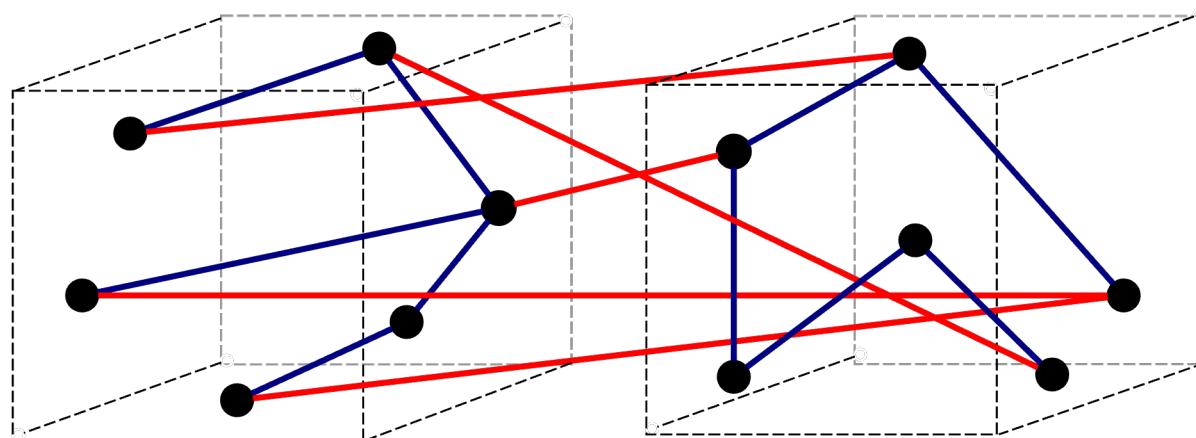
51160572100266670029129

3





a) Good (loose coupling, high cohesion)



b) Bad (high coupling, low cohesion)

TO VANLIGE MODELLERINGSFEIL

#1 Å PRØVE Å FINNE ELEFANTENS SANNE NATUR

**#2 Å TRO AT DET BARE
FINNES ÉN MODELL AV
DOMENET**

**MODELLER ER IKKE
SANNE ELLER GALE**

**MODELLER ER
NYTTIGE FORENKLINGER**

**MODELLER ER
MER ELLER MINDRE EGNET**

**MODELLER ER
MER ELLER MINDRE NYTTIGE**

MODELLER ER LOKALE

UBIQUITOUS LANGUAGE

**FELLES OG ENTYDIG SPRÅK
INNEN EN BOUNDED CONTEXT**

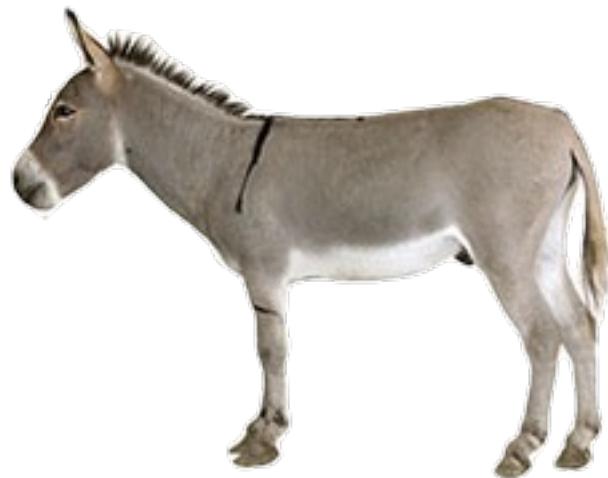
HVORFOR ER SPRÅK SÅ VIKTIG?

HISTORIEN OM EQUEST

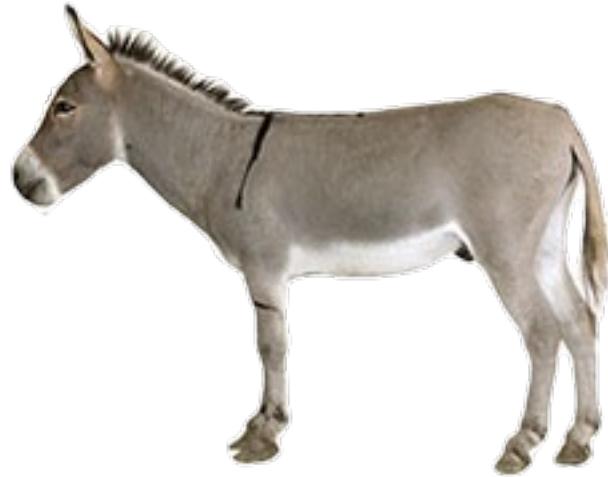
EN HEST



EN NY HEST



EN KORT, STERK OG STA HEST



EN HEST?

EN HEST?



EN HEST?



+



EN VANLIG HEST



EN ALMINNELIG HEST



EN TYPISK HEST



EN EKTE HEST



EN GAMMELDAGS HEST



EN OPPRINNELIG HEST



EN HESTE-HEST



```
if (horse.IsShort && horse.IsStubborn) {  
    // Logic for the new horse case.  
}  
else  
{  
    // Regular horse code here.  
}
```

BUGS!

```
if (horse.IsShort &&
    horse.IsStubborn &&
    horse.Sound == Sound.HeeHaw)
{
    // Logic for the new horse case.
}
else
{
    // Regular horse code here.
}
```

EN NY NY HEST



**EN SÅNN HEST SOM ER AVKOMMET
AV EN VANLIG HEST-HEST**



**OG EN KORT OG STERK OG STA HEST
SOM SIER HI-HÅÅ**

```
if (horse.IsShort &&
    horse.IsStubborn &&
    horse.Sound == Sound.HeeHaw) ||
(horse.Sire.IsShort &&
 horse.Sire.IsStubborn &&
 horse.Sire.Sound == Sound.HeeHaw) ||
(horse.Dam.IsShort &&
 horse.Dam.IsStubborn &&
 horse.Dam.Sound == Sound.HeeHaw))
{
    // Logic for both the new horse
    // and the new-new horse!
} else
{
    // Really regular horse code here.
}
```

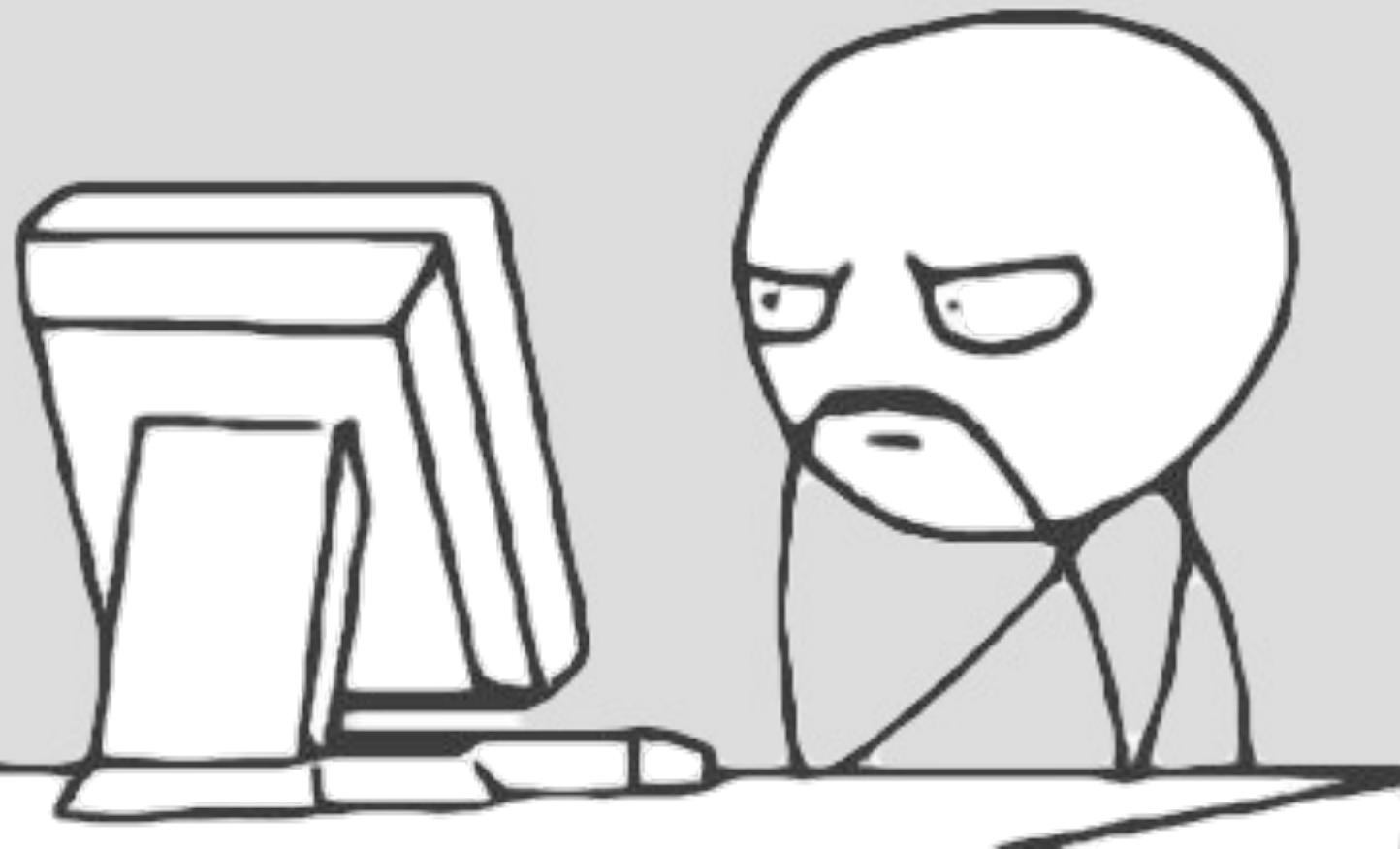
BUGS!

ER DET MOREN ELLER FAREN TIL DENNE HESTEN



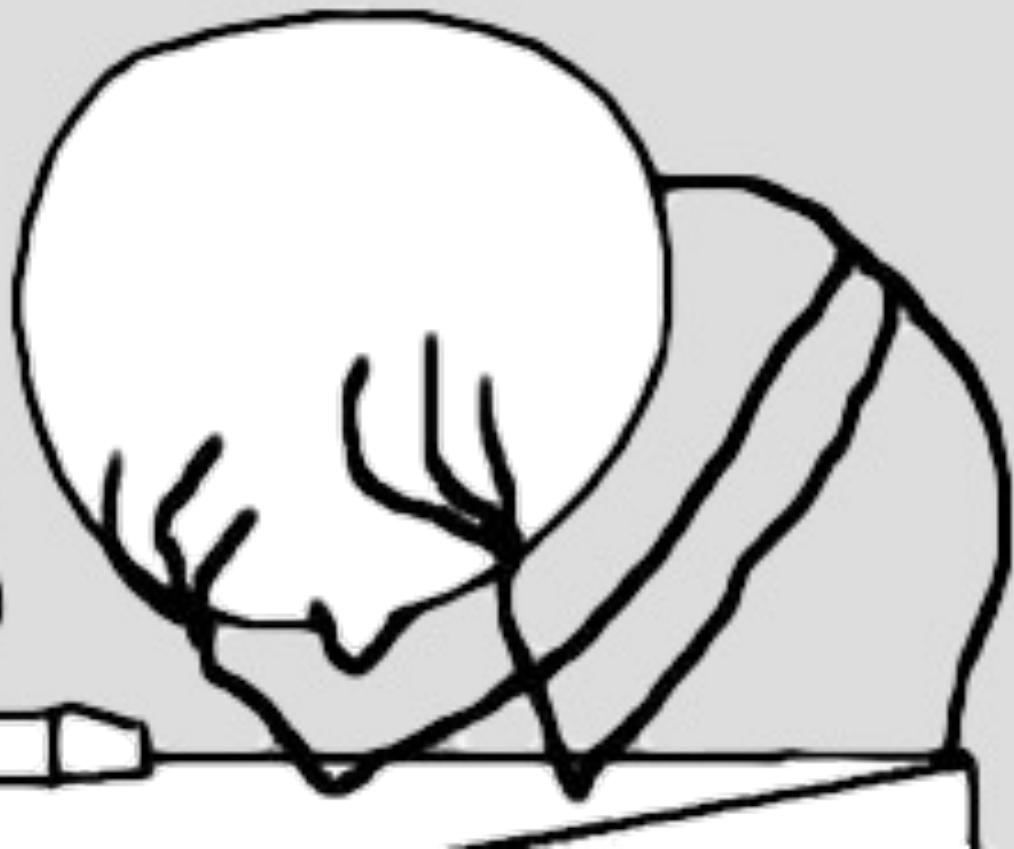
**SOM ER KORT OG STERK OG STA
OG SIER HI-HÅÅ?**

«TEKNISK GJELD»



EN NY NY NY HEST





SPRÅK ER VIKTIG!

SPRÅK ER VANSKELIG!

FELLES SPRÅK FOR TEAMET

FELLES SPRÅK FOR DET KRYSSFUNKSJONELLE TEAMET

UNNGÅ «STAMMESPRÅK»

UNNGÅ OVERSETTELSER

UNNGÅ FLERTYDIGHETER

UNNGÅ SKINN-ENIGHETER

**DETTE ER SKIKKELIG VIKTIG
OG SKIKKELIG VANSKELIG**

**DET ER VANSKELIG Å ENDRE
PÅ EGEN SPRÅKBRUK**

**SPRÅK ER DET
VI TENKER MED**

**VI MÅ KUNNE ENDRE PÅ
BYGGEKLOSSENE I HODET**

**ALLE PÅ TEAMET MÅ VÆRE
BEVISST PÅ SPRÅKBRUKEN**

VÅRE MENTALE MODELLER
BYGGES OPP AV
DET FELLES SPRÅKET

**BRUKSTILFELLER
SKAL UTTRYKKES I
DET FELLES SPRÅKET**

**KODEN SKAL BRUKE
BEGREPER FRA
DET FELLES SPRÅKET**

ET LEVENDE SPRÅK

EN LEVENDE MENTAL MODELL

VÆR PÅ UTKIKK
ETTER STEDER DER
DISKUSJONEN LUGGER!

**FRUSTRASJON OG IRRITASJON
ER SYMPTOMER PÅ ET
MANGELFULLT SPRÅK**

HVORFOR BOUNDED?

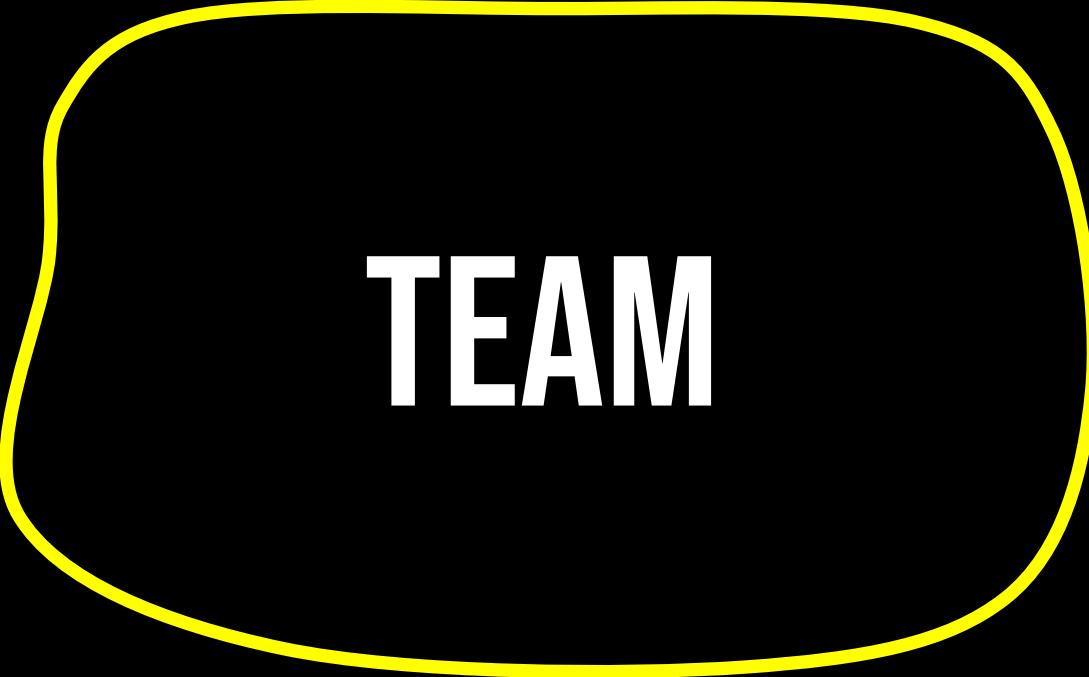
FORMÅLET MED AVGRENSNING

FLERE KONTEKSTER

KOGNITIV LAST

KOMMUNIKASJON

**BOUNDED
CONTEXT**



TEAM



AUTONOMI

ddd-crew/bounded-context-canvas

Name:	V5 github.com/ddd-crew/bounded-context-canvas																						
Purpose What benefits does this context provide, and how does it provide them? Describe the purpose from a business perspective	Strategic Classification <table> <tr> <td>Domain</td> <td>Business Model</td> <td>Evolution</td> </tr> <tr> <td>- core</td> <td>- revenue</td> <td>- genesis</td> </tr> <tr> <td>- supporting</td> <td>- engagement</td> <td>- custom built</td> </tr> <tr> <td>- generic</td> <td>- compliance</td> <td>- product</td> </tr> <tr> <td>- other?</td> <td>- cost reduction</td> <td>- commodity</td> </tr> </table>	Domain	Business Model	Evolution	- core	- revenue	- genesis	- supporting	- engagement	- custom built	- generic	- compliance	- product	- other?	- cost reduction	- commodity	Domain Roles <table> <tr> <td>Role Types</td> </tr> <tr> <td>- draft context</td> </tr> <tr> <td>- execution context</td> </tr> <tr> <td>- analysis context</td> </tr> <tr> <td>- gateway context</td> </tr> <tr> <td>- other</td> </tr> </table>	Role Types	- draft context	- execution context	- analysis context	- gateway context	- other
Domain	Business Model	Evolution																					
- core	- revenue	- genesis																					
- supporting	- engagement	- custom built																					
- generic	- compliance	- product																					
- other?	- cost reduction	- commodity																					
Role Types																							
- draft context																							
- execution context																							
- analysis context																							
- gateway context																							
- other																							
Inbound Communication <table> <tr> <th>Collaborator</th> <th>Messages</th> </tr> <tr> <td></td> <td>  </td> </tr> </table>	Collaborator	Messages			Ubiquitous Language Context-specific domain terminology 	Outbound Communication <table> <tr> <th>Messages</th> <th>Collaborator</th> </tr> <tr> <td>  </td> <td></td> </tr> </table>	Messages	Collaborator															
Collaborator	Messages																						
																							
Messages	Collaborator																						
																							
Assumptions Describe which currently unverified assumptions went into this bounded context design. Make those assumptions explicit by documenting them here	Verification Metrics Describe metrics which can be used to (in)validate the current structure of this bounded context?	Open Questions																					

HVA SKJER VED GRENSEN?

**HVORDAN FORHOLDER
KONTEKSTER SEG
TIL HVERANDRE?**

CONTEXT MAP

EKSPLISITTE GRENSESTRATEGIER

PROBLEMSTILLINGER

HVOR STOR GRAD AV
OVERLAPP ER DET MELLOM
MODELLENE?

HVA ER STYRKEFORHOLDET MELLOM TEAMENE?

**HVOR STOR INNFLYTELSE
HAR MAN PÅ
DEN ANDRE MODELLEN?**

**HVOR STOR INNFLYTELSE
HAR MAN PÅ
DET ANDRE TEAMET?**

EKSEMPEL PÅ STRATEGI

ANTI-CORRUPTION LAYER

ISOLERENDE LAG MELLOM EGEN KONTEKST OG EKSTERN KONTEKST

**BESKYTTE MODELLEN OG
SPRÅKET MOT EKSTERN
PÅVIRKNING**

**UNNGÅ AT ANDRES BEGREPER
OG HENSYN LEKKER INN
I EGEN KONTEKST**

HVORFOR ISOLERE?

INTEGRITET OG SELVSTENDIGHET

**TRENGER IKKE Å VÆRE NOE
GALT MED DEN ANDRE
KONTEKSTEN**

**DEN ANDRE KONTEKSTEN
FOKUSERER PÅ Å LØSE
ET ANNET PROBLEM**

HVORFOR ELLERS ER DET TO KONTEKSTER OG TO MODELLER?

EKSEMPEL

SILJE-PROSJEKTET FOR MATTILSYNET

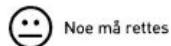
Tilsynsresultat

FOOD HYGIENE RATING

Steffi Støvo
FJELL
5357 FJELL



Tilsynsdato: 25.06.2016



DETTE HAR MATTILSYNET SETT PÅ:

Ledelse og rutiner

Ikke funnet regelverksbrudd som krever oppfølging.

Lokaler og utstyr

Ikke funnet regelverksbrudd som krever oppfølging.

Mathåndtering og tilberedning

Ikke funnet regelverksbrudd som krever oppfølging.

Sporbarhet og merking

Ikke funnet regelverksbrudd som krever oppfølging.

TIDLIGERE TILSYNSRESULTATER:

Mattilsynet har kontrollert etterlevelsen av sentrale krav i matlovgivningen. Resultatene baserer seg på observasjonene som ble gjort og de opplysningene vi fikk under inspeksjonen.

For mer detaljert informasjon, gå til
www.matportalen.no/smilefjes



SILJE OG MATS

SILJE SMILEFJESTILSYN

SILJE LITE OG SPESIFIKT

MATS SAKSBEHANDLINGSSYSTEM

MATS
STORT OG GENERISK

**SILJE MÅ INTEGRERE
MOT MATS**

**SILJE MÅ BESKYTTE SEG
MOT DÅRLIG INNFLYTELSE
FRA MATS**

SILJE

J
E
N
S

MATS

SILJE

A
C
L

J
E
N
S

MATS

SILJE

A
C
L

J
E
N
S

MATS

**SPRÅK OG MODELL I SILJE
BLE LANGT ENKLERE OG
MER PRESIST ENN I MATS**

**SILJE BLE MER FLEKSIBELT
OG ENKLERE Å ENDRE**

KONSEKVENSER HELT OPP I SKJERMBILDENE

BEDRE BRUKEROPPLEVELSE

SUNNERE KODEBASE

FLERE GRENSESTRATEGIER

Context Map Cheat Sheet

Context Map Patterns

Open / Host Service

A Bounded Context offers a defined set of services that expose functionality for other systems. Any downstream system can then implement their own integration. This is especially useful for integration requirements with many other systems. Example: public APIs.



Conformist

The downstream team conforms to the model of the upstream team. There is no translation of models. Couples the Conformist's domain model to another bounded context's model.



Anticorruption Layer

The antิcorruption layer is a layer that isolates a client's model from another system's model by translation. Only couples the integration layer (or adapter) to another bounded context's model but not the domain model itself.



Shared Kernel

Two teams share a subset of the domain model including code and maybe the database. Typical examples: shared JARs, DLLs or a shared database schema. Teams with a Shared Kernel are often mutually dependent and should form a Partnership.



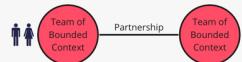
Customer / Supplier

There is a customer / supplier relationship between teams. The downstream team is considered to be the customer. Downstream requirements factor into upstream planning. Therefore, the downstream team gains some influence over the priorities and tasks of the upstream team.



Partnership

Partnership is a cooperative relationship between two teams. These teams establish a process for coordinated planning of development and joint management of integration.



Published Language

A Published Language is a well documented shared language between Bounded Contexts which can translate in and out from that language. Published Language is often combined with Open Host Service. Typical examples are iCalendar or vCard.



Separate Ways

Bounded Contexts and their corresponding teams have no connections because integration is sometimes too expensive or it takes very long to implement. The teams chose to go separate ways in order to focus on their specific solutions.



Big Ball Of Mud

A (part of a) system which is a mess by having mixed models and inconsistent boundaries. Don't let this lousy model propagate into the other Bounded Contexts. Big Ball Of Mud is a demarcation of a bad model or system quality.



Team Relationships

Mutually Dependent

Two software artifacts or systems in two bounded contexts need to be delivered together to be successful and work. There is often a close, reciprocal link between data and functions between the two systems.



Free

Changes in one bounded context do not influence success or failure in other bounded contexts. There is, therefore, no organizational or technical link of any kind between the teams.



Upstream / Downstream

Actions of an upstream team will influence the downstream counterpart while the opposite might not be true. This influence can apply to code but also on less technical factors such as schedule or responsiveness to external requests.



HVILKEN STRATEGI ER RIKTIG?

DET KOMMER AN PÅ!

KONTRAKT MED OMVERDENEN

SAMARBEIDSFORM

KOMMUNIKASJONSNIVÅ

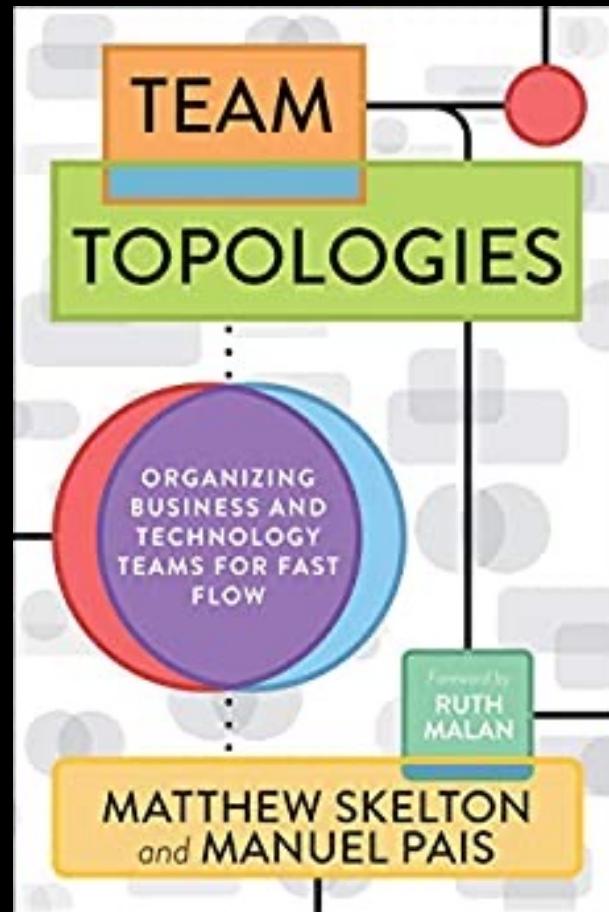
AUTONOMITETSGRAD

TEAM

BOUNDED CONTEXT SOM TEAMGRENSE

LA DEN STRATEGISKE PARTISJONERINGEN AV DOMENET DRIVE TEAM-INNDELINGEN

TEAM TOPOLOGIES



DOMENEDREVET ORGANISASJONSDESIGN

CONWAYS LOV

OPTIMALISERINGSPROBLEM

KNAPPHET

KOGNITIV LAST

KOMMUNIKASJON

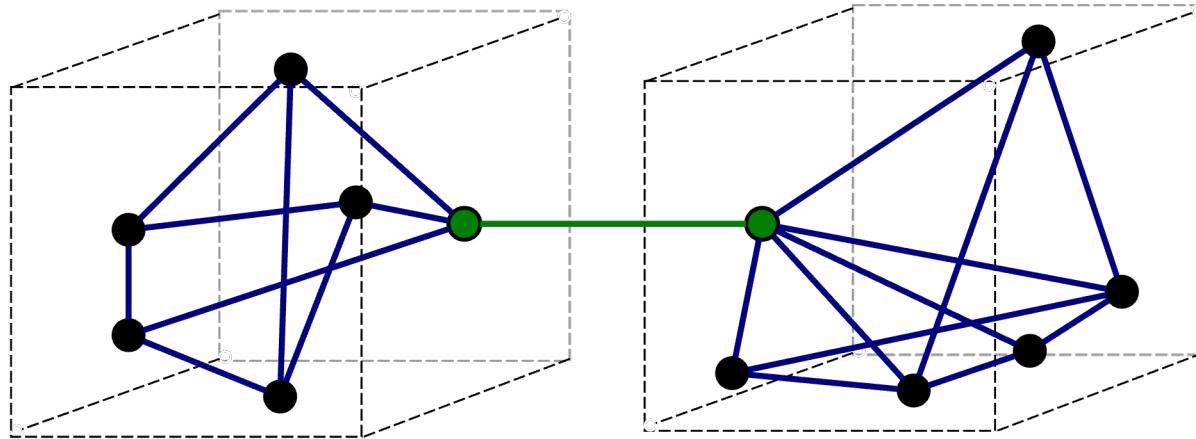
KOMMUNIKASJON SOM BEGRENSET RESSURS

KOMMUNIKASJON INTERNT I TEAMET

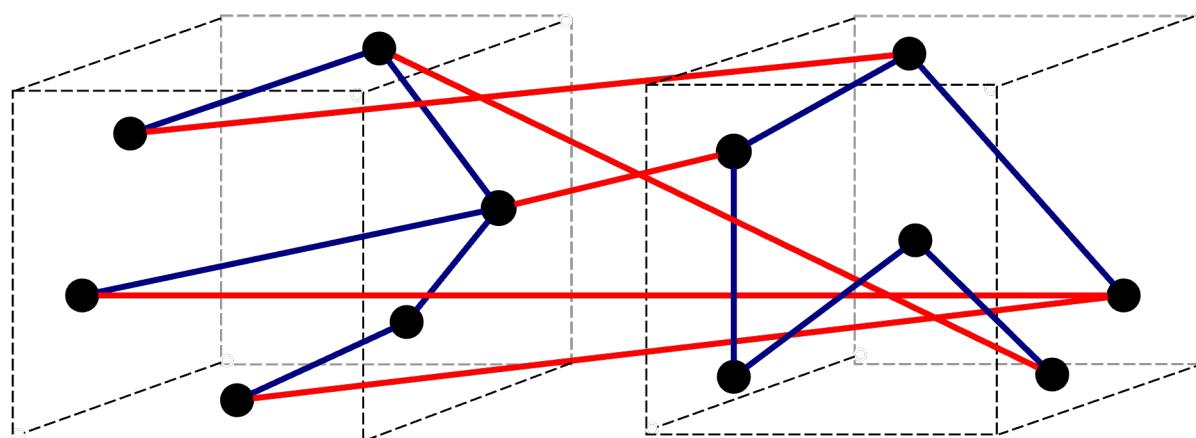
KOMMUNIKASJON MELLOM TEAM

KOMMUNIKASJON MELLOM ORGANISASJONER

**HIGH COHESION
LOW COUPLING**



a) Good (loose coupling, high cohesion)



b) Bad (high coupling, low cohesion)

VERDISTRØMMER

HVORDAN ANVENDE TEAM TOPOLOGIES?

«INVERSE CONWAY»

HA HA!

HOMOMORPHIC FORCE

SOSIALT



TEKNISK

KOMMUNIKASJON



PRODUKSJON

**REORGANISERING
ER IKKE NOK**

**THE GHOST OF THE OLD
SYSTEM CONTINUES TO
HAUNT THE NEW**

SOSIOTEKNISK REFAKTORERING

GJENSIDIG OPTIMALISERING

ENDRE I TAKT

HVEM MÅ VÆRE INVOLVERT?

UTENFOR TEAMETS AUTONOMI



MINST TO TEAM

+ OMGIVELSEN

KOSTBART

RISIKO

GEWINST

FLYTTE EN DOMENEGRNSE

FRA SYNC TIL ASYNC

OVERFØRE ET SYSTEM

SKRU AV ET SYSTEM

KUTTE EN AVHENGIGHET

INNFØRE REDUNDANS

FRA SAMARBEID TIL SELVBETJENING

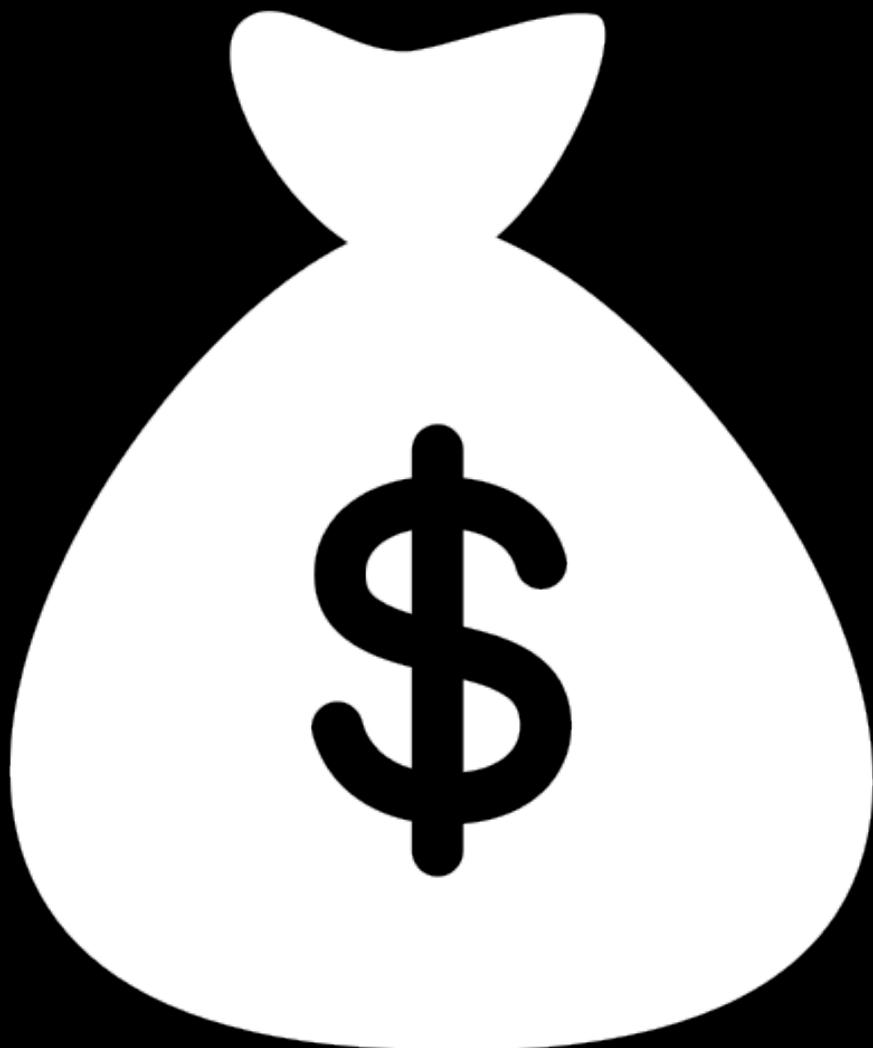
REDUSERE KOSTNADEN VED KOMMUNIKASJON

ØKE COHESION
REDUSERE COUPLING

KOMPLISERENDE FAKTORER

PRIORITET

**HOLDES IGJEN AV
ANDRE STRUKTURER**



HVEM KAN ENDRE PÅ DE STRUKTURENE?

HVEM ER ARKITEKTEN NÅ?

HVA ER AUTONOMI?

**PROGRAMVAREUTVIKLING ER
SOSIOTEKNISK**

**VI BYGGER IKKE
BARE PRODUKTET**

**VI BYGGER OGSÅ
ORGANISASJONEN SOM
BYGGER PRODUKTET**

BÆREKRAFTIGE OG ENDRINGSODYKTIGE SOSIOTEKNISKE SYSTEMER