Slik lykkes du med arkitektur i en smidig verden

Webinar, 17. juni 2020, David Skogan, david.skogan@promis.no, +4741649156

PROMIS
Project Management • Information Systems

1

Agenda

- Bakgrunn
- Endringstakt og smidige utvikling
- Opplevd teameffektivitet
- Årsaker
  - Avhengigheter (for mange)
  - Organisering
  - Arkitektur
- Arkitekturprinsippet: Low Coupling
- Noen råd

PROMIS
Project Management • Information Systems

2

Bakgrunn

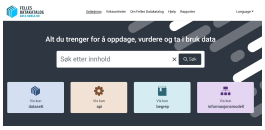
PROMIS
Project Management • Information Systems

ITPROJECTPROFESSIONAL

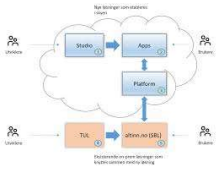
ACE

Løsningsarkitekt
Java / C#
Smidig utvikling

Felles datakatalog



Altinn Tjenester 3.0

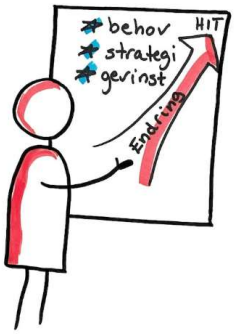


PROMIS
Project Management • Information Systems

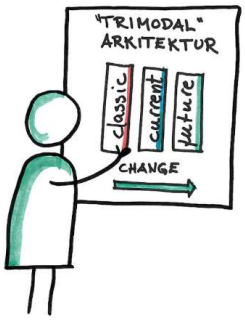
PROMIS arkitekturtjenester

Informasjonsforvaltning

Digital transformasjon



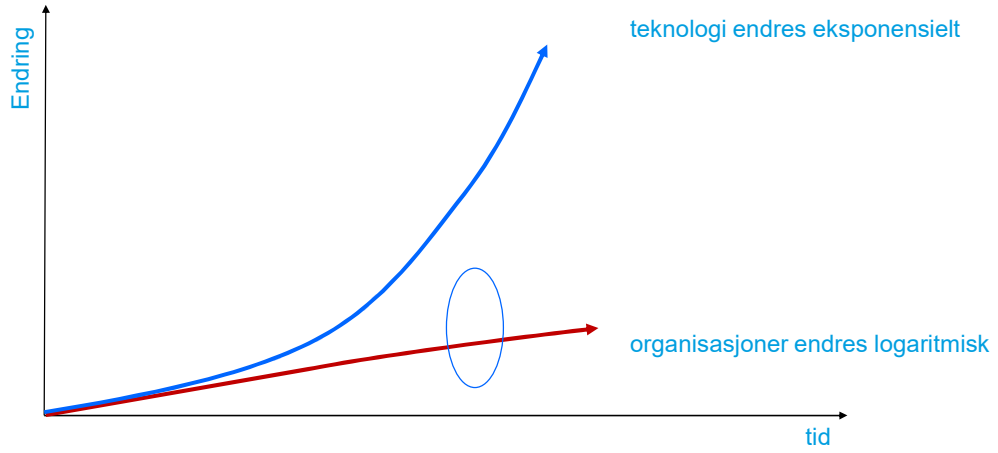
Arkitekturutvikling



PROMIS
Project Management • Information Systems

Endringstakt

Martec's Law (Scott Brinker)



PROMIS Project Management • Information Systems

5

Kjennetegn ved IT-prosjekter som lykkes (SMIOS-prosjektet 2018/SIMULA)

- God kontroll på ambisjonsnivå. Unngår "for mye" på en gang og sier nei til tillegg som vil øke risiko.
- Bruker kontrakter som unngår "fastpris-oppførsel" for leverandør (og kunde): Per time eller målpris uten tak
- Har en kunde (bestiller-team) med kompetanse til å velge og styre kompetente leverandører og enkeltressurser (og som har mindre fokus på pris).
- Har fleksibilitet i omfang (ikke bare må-funksjonalitet)
- Har en kunde som (minimum) er sterkt involvert i planlegging av nytte og nyttestyring underveis
- **Bruker smidig utvikling med hyppige leveranser til produksjon** (eller i det minste til realistisk evaluering)
  - Støttes av gode prosesser for effektiv testing
- Tenker tidlig på (og involverer) overlevering til linje, brukere og drift

PROMIS Project Management • Information Systems

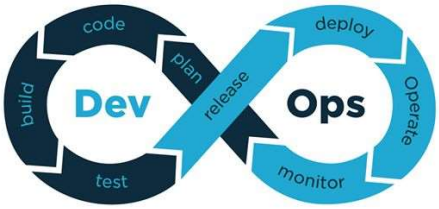
6

Smidig utvikling (siden 2001)

Scrum

Kanban

XP



PROMIS Project Management • Information Systems

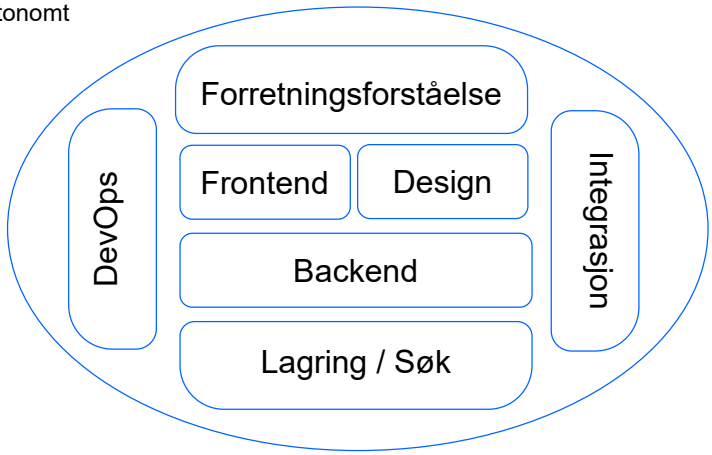
7

Smidig team – maksimere fleksibilitet, kreativitet og produktivitet

- Selvorganiserende og autonomt
- Tverrfaglig

Produkteier

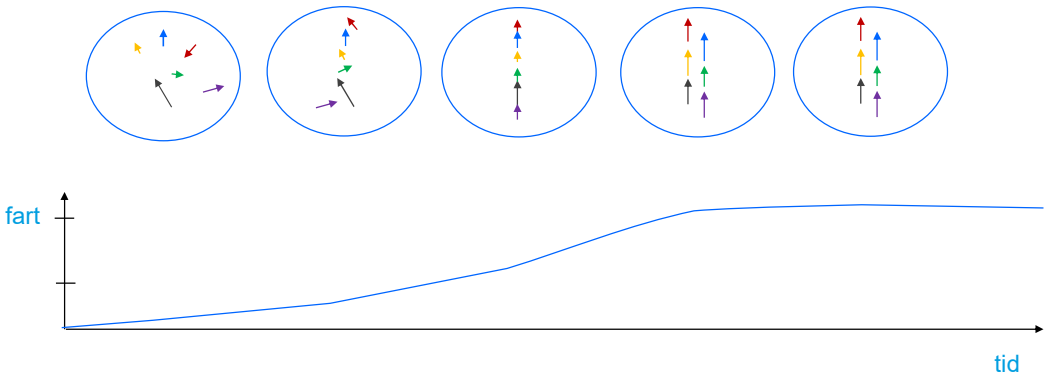
Scrum master/
Agile coach



PROMIS Project Management • Information Systems

8

Opplevd effektivitet: Fart og endringstakt (lead time)



PROMIS Project Management • Information Systems

9

Arkitekt

Som arkitekt ønsker jeg å ta arkitekturvalg som gjør at produktet oppfyller kravene til funksjonalitet og ytelse, og ***samtidig gjør det er enkelt og kostnadseffektivt å utvikle og vedlikeholde systemet nå og i fremtiden****.*

Som arkitekt ønsker jeg tilbakemelding på arkitekturvalg slik at man kan avgjøre hvilke valg som gir mest effekt.

Arkitektur må kommuniseres, forstås og utvikles

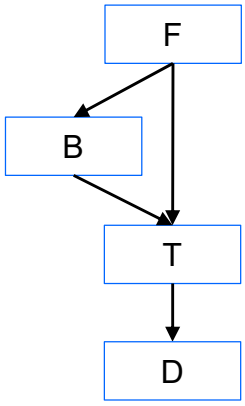
PROMIS Project Management • Information Systems

10

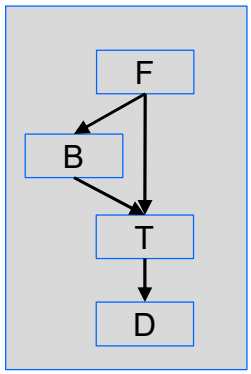
Arkitektur: Fra logisk til fysisk

Architectural Quanta

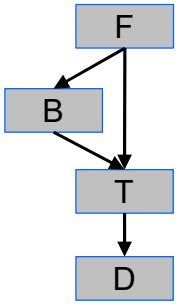
Logisk arkitektur



Fysisk arkitektur
Monolitt



Fysisk arkitektur
Mikrotjenester

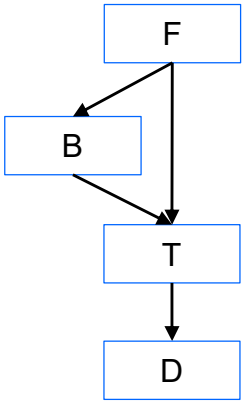


PROMIS
Project Management • Information Systems

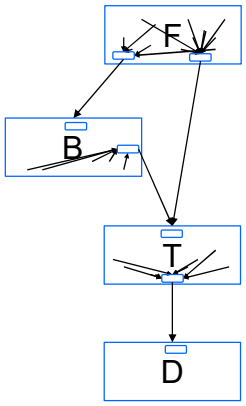
11

Koblinger i kode

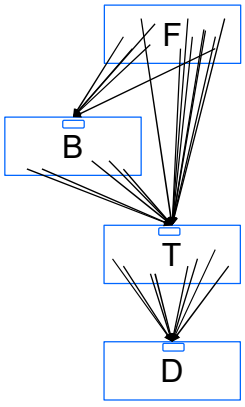
Logisk arkitektur



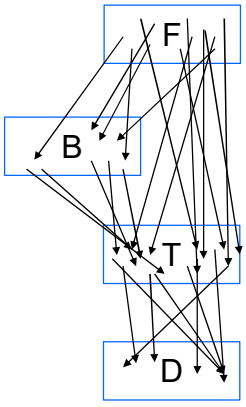
Løst koblet



Løsere koblet



Tett koblet

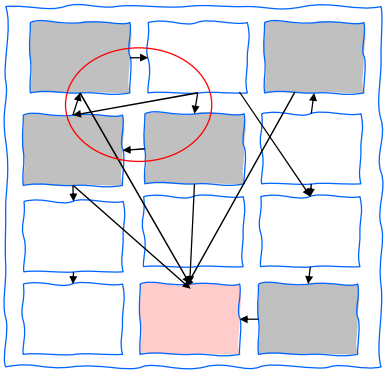


PROMIS
Project Management • Information Systems

12

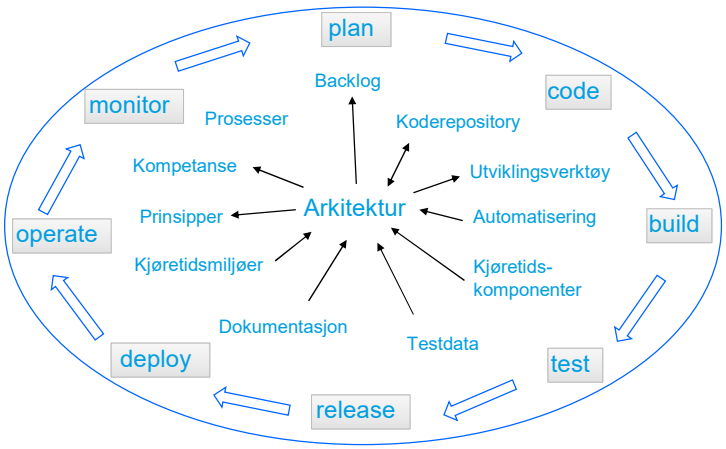
Arkitekturprinsipp: Low Coupling (Larry Constantine, 70-tallet)

- Low Coupling – High Cohesion
- Tett koblede moduler er ikke bra
  - Endring i en modul kan kreve at avhengige moduler også endres
  - En modul er vanskelig å gjenbruke/teste fordi avhengige moduler må inkluderes
  - Vanskelig å bryte opp i mindre mer løst koblede moduler
- Løs opp avhengighetene!
  - Dependency inversion fra SOLID prinsippene
  - Mikrotjenester



Hvilke andre avhengigheter har vi?

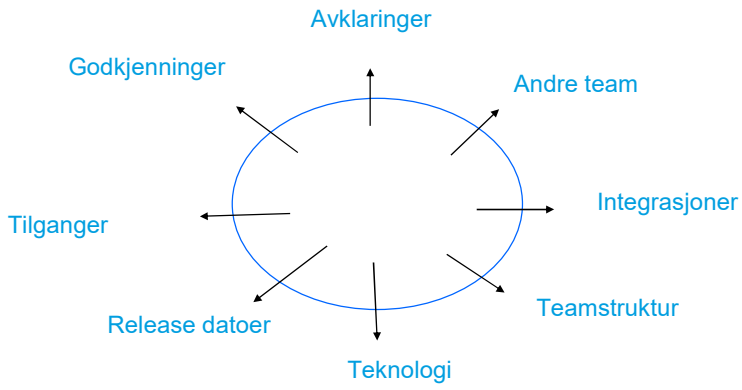
Interne avhengigheter



PROMIS
Project Management • Information Systems

15

Eksterne avhengigheter



PROMIS
Project Management • Information Systems

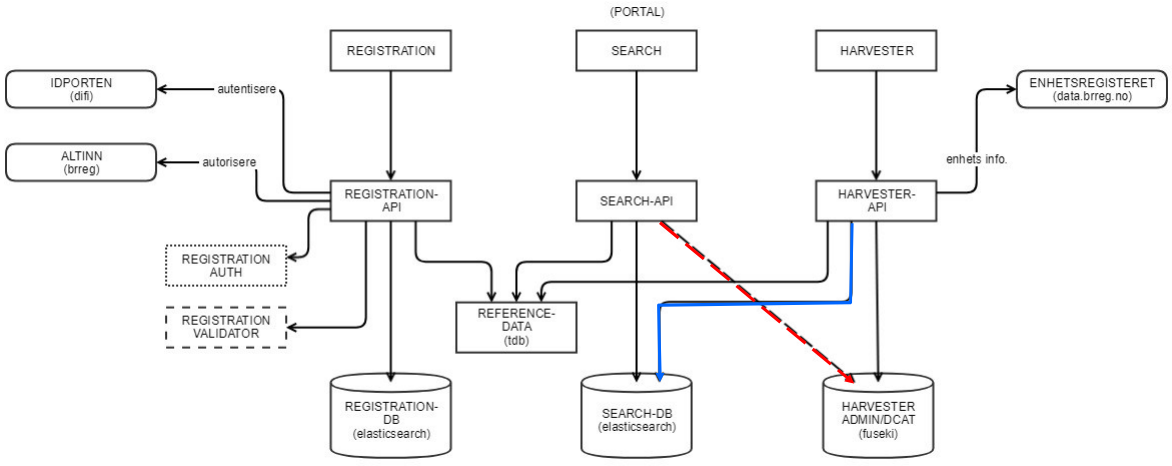
16

Hvordan løse opp i avhengigheter

- Up front arkitektur!
- Up front organisering!
- Identifiser avhengigheter som hindrer oss
  - Mål/dokumenter
  - Lag plan og gjør endringer
  - Automatiser
- Bidrar arkitekturen til den ønsket endringstakt?
  - Har vi riktig størrelse på komponentene (Architectural Quanta)

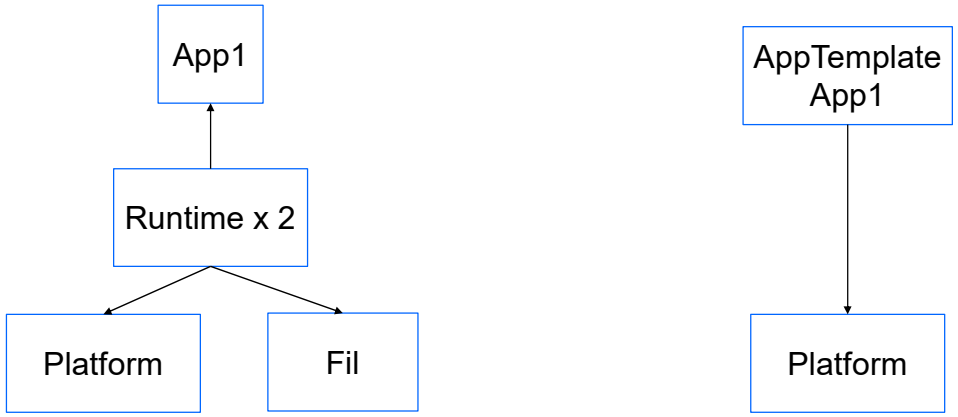
17

Identifiser avhengigheter og lag en plan

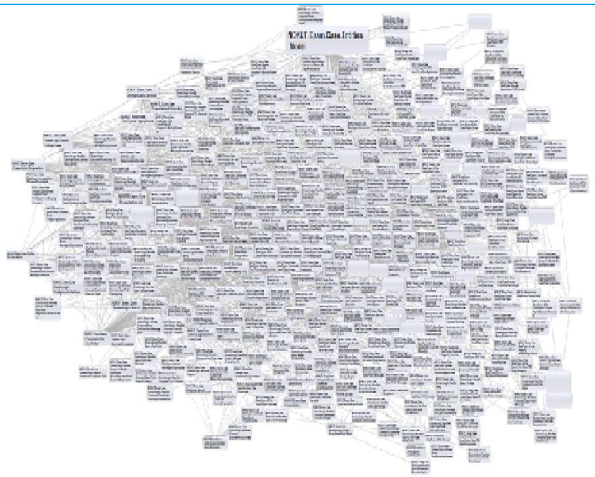


18

Endre arkitekturen hvis du ikke får testet koden



Arkitektur på flere nivå – Bruk verktøy!



400+ navnerom/moduler

Teknologi spesifikk inndeling ☹

Tegn på dårlig arkitektur?

- Tegningene henger ikke fremme
- Stemmer ikke med hva som bygges
- Lav utviklingsfart
- Antall feil øker
- Utdatert teknologi, gamle versjoner av rammeverk
- Lav testdekning
- Utviklere slutter

Vanskelig å gjøre endringer!

PROMIS Project Management • Information Systems

21

Tegn på god arkitektur

- Arkitekturtegningene henger på veggen
- Blir aktivt brukt og oppdatert
- Bidrar til teamorganisering
- Kontroll på avhengigheter
- Lett å onboarde nye utviklere
- God testdekning og høy grad av automatisering
- Høy utviklingstakt

Enkelt å gjøre endringer!

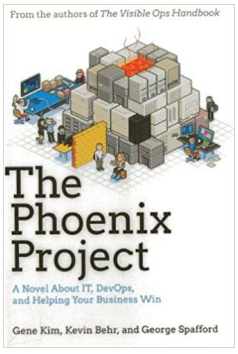
PROMIS Project Management • Information Systems

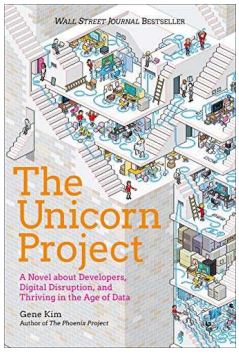
22

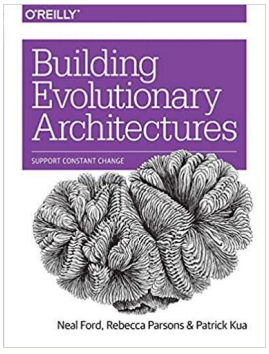
Så hvordan lykkes du med arkitektur i en smidig verden?

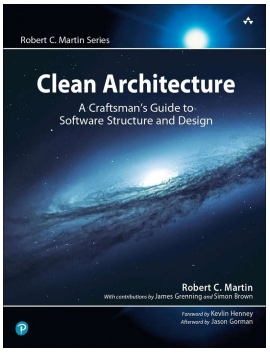
- Få kontroll på avhengighetene dine
  - Vi må være i stand til å gjøre endringer raskt og uten å påvirke resten av systemet
  - Low Coupling – Færre avhengigheter
  - Bruk verktøy til å måle og finne avhengigheter
- Noen råd
  - Noe up-front arkitektur/organisering
  - Endringer i arkitektur kan bidra til enklere arbeidsfordeling og bedre effektivitet
  - Arkitekturen må utvikles etter hvert som vi lærer (evolusjonær arkitektur)
  - Mål fart og endringstakt
  - Korte sprinter
  - Arkitektur-review

Litteratur










Best practice
Managing uncertainty
Increasing efficiency
Integrated co-operation
www.promis.no
Agile projects
Best Practice
Integrated co-operation

25