IN 2001

29 januar 2018

Teamarbeid og smidig metodikk. Lean og Scrum. Prosjektarbeid

UNIVERSITETET I OSLO

Yngve Lindsjørn

ynglin@ifi.uio.no

Utvikling av programvare - Suksesskriterier

- Levere programvaren til rett tid
- Holde kostnadene innen budsjett
- Levere programvare som møter kundens forventninger og krav
- Opprettholde et velfungerende utviklingsteam

Ulike typer av risiko

Risiko type	Mulige risikoer
Teknologi	Databasen som brukes i systemet klarer ikke å prosessere så mange transaksjoner per sekund som forventet
Mennesker	Umulig å rekruttere mennesker med den kompetanse som kreves Nøkkel personell ikke tilgjengelig i kritiske faser
Organisasjon	Organisasjonen blir restrukturert slik at ulik ledelse er ansvarlig for prosjektet
Verktøy	Ulike programvareverktøy lar seg ikke integrere
Krav	Endringer av krav krever omfattende "redesign”
Estimering	Underestimering (i tid) av programvareutviklingen Tiden det tar å rette feil er underestimert

Risikotyper og eksempler

Risiko	Sannsynlighet	Konsekvens
Det er umulig å rekruttere medarbeidere med kompetansen som er nødvendig	Høy	Katastrofal
Nøkkel personell er syke eller fraværende i kritiske faser av prosjektet	Moderat	Alvorlig
Det er foreslått endringer i kravspesifikasjonen som vil kreve store endringer i design av systemet	Moderat	Alvorlig
Organisasjonen restruktureres slik at ulik ledelse har ansvar for prosjektet	Høy	Alvorlig
Databasesystemet kan ikke prosessere antall transaksjoner per sekund som forventet	Moderat	Alvorlig

Teamarbeid

- Et team som fungerer er samlet og har en god teamfølelse.
- Teamets mål viktigere enn egne mål
- Kommunikasjon er en nøkkelfaktor for å lykkes
- Fleksibilitet i teamsammensetning er ofte begrenset av hvem som er tilgjengelig

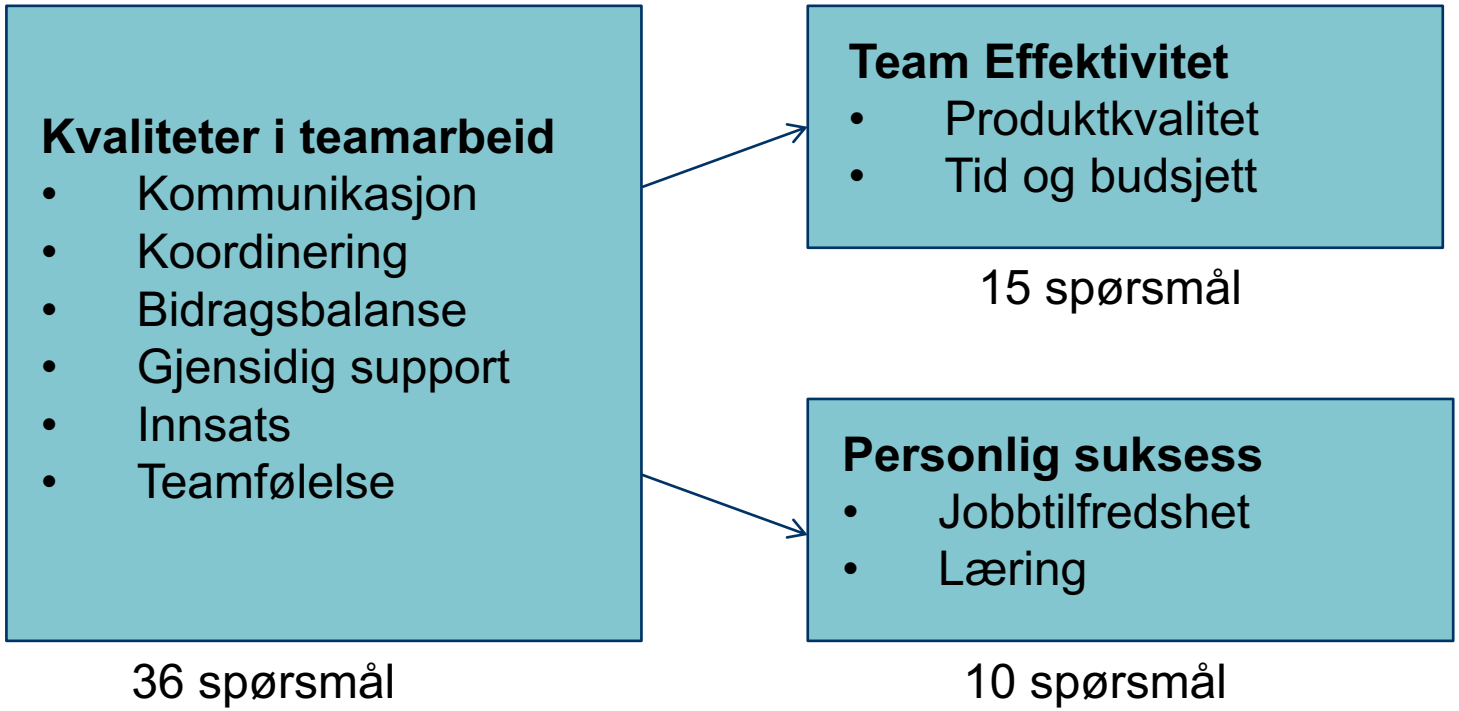
Hva er spesielt med Software-team?

- Hyppige endringer gjør det vanskelig å planlegge
- Komplekse sosiale og tekniske system
- Få etablerte teorier om systemutvikling

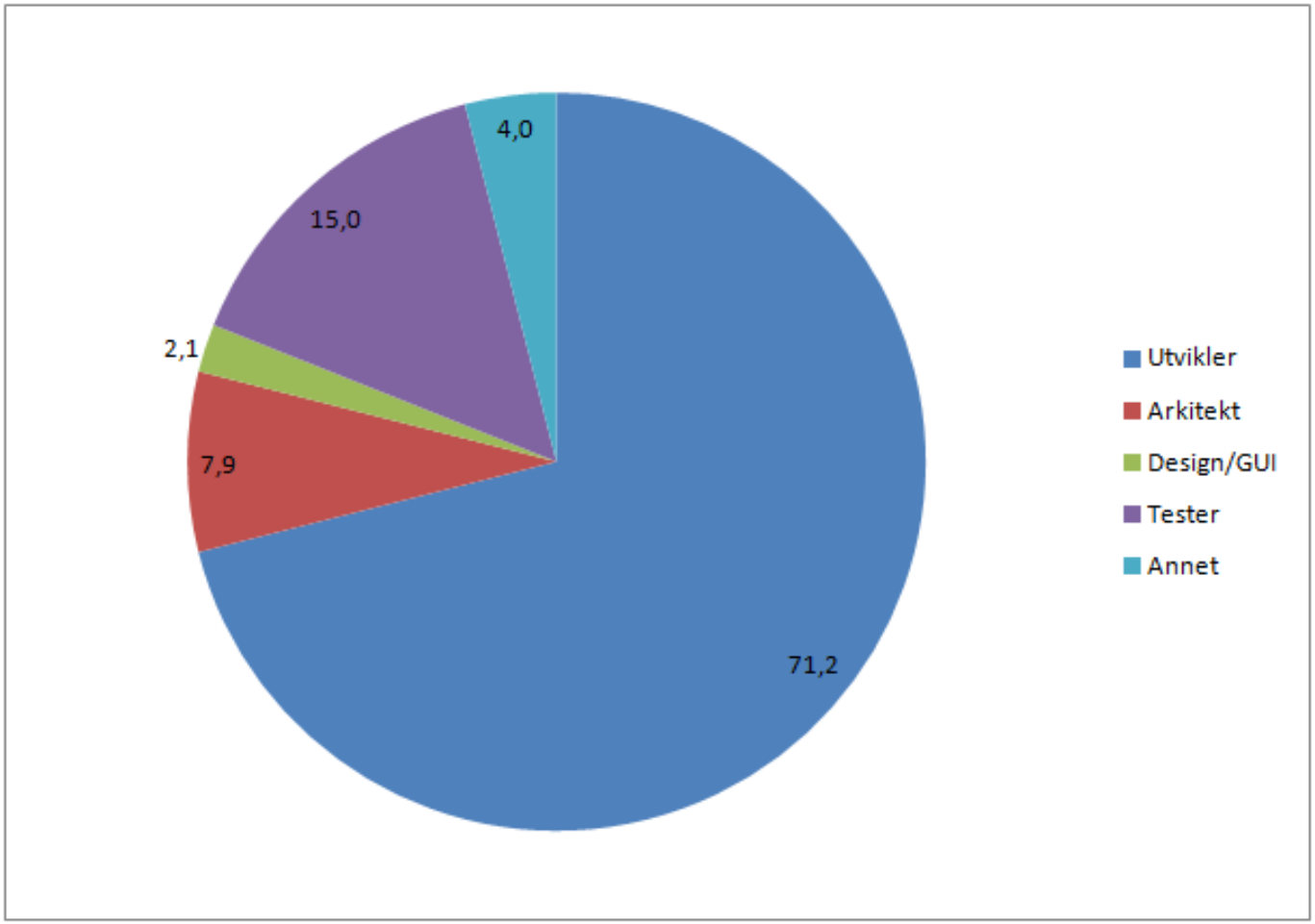
Effektivitet i team

- Teammedlemmer
  - Trenger en blanding av folk fordi systemutvikling involverer ulike aktiviteter som programmering, testing, programvarearkitektur og dokumentasjon
- Teamorganisering
  - Et team bør organiseres slik at alle teammedlemmene kan bidra best mulig og at oppgaver blir utført som forventet
- Kommunikasjon
  - God kommunikasjon mellom teammedlemmene, og mellom teamet og andre interessenter (stakeholders), er helt essensielt

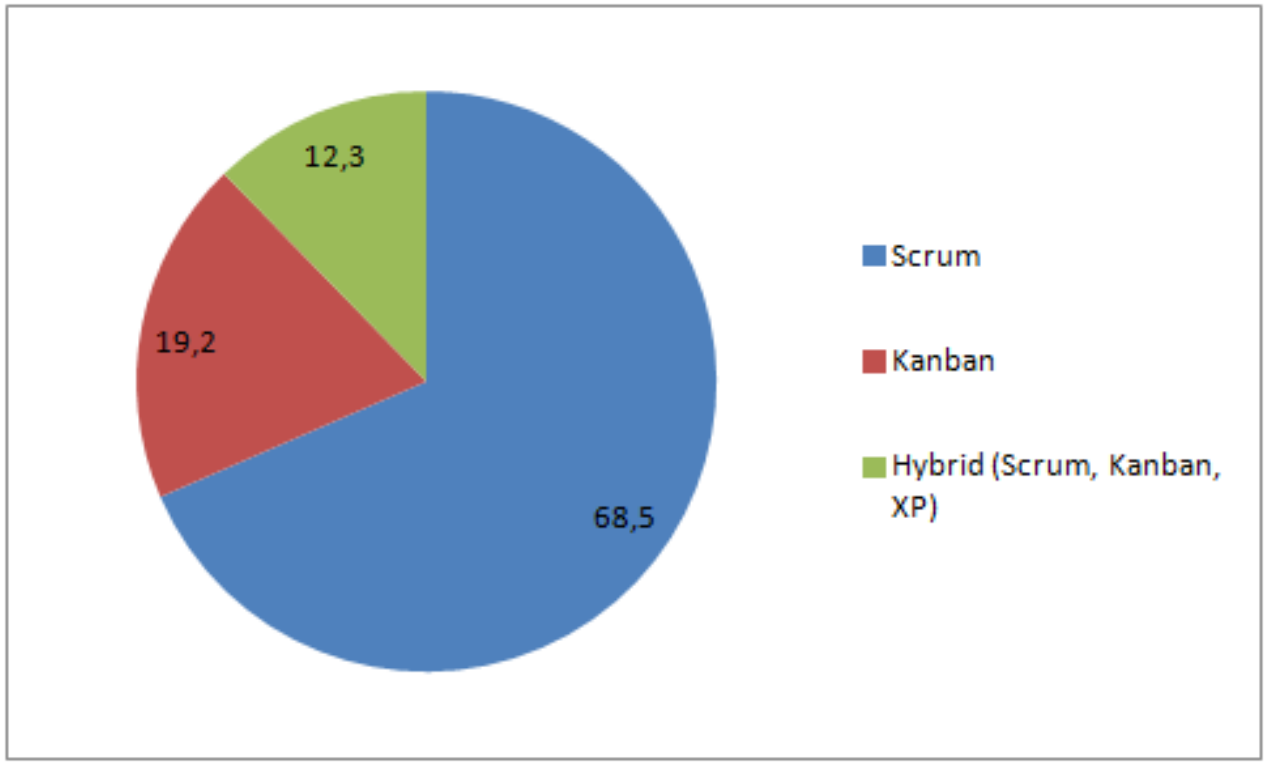
Teameffektivitets-modell i studie



Teammedlemmer – primærrolle i team



Type ”smidig” team



Smidig planlegging

- Smidige metoder i programvareutvikling er en iterativ tilnærming der programvaren blir utviklet og levert til kundene som ”tillegg” (”increments”)
- Til forskjell for plandrevet tilnærming, er funksjonaliteten til tilleggene (increments) ikke planlagt på forhånd men avgjøres under utviklingen
  - Hva som tas med i en iterasjon avhenger av utvikling i prosjektet og kundens prioriteringer
- Kundens prioriteringer og krav endrer seg. Derfor kan det være fornuftig å ha en fleksibel plan som kan ta høyde for disse endringene

Historiebasert (”Story-based”) planlegging

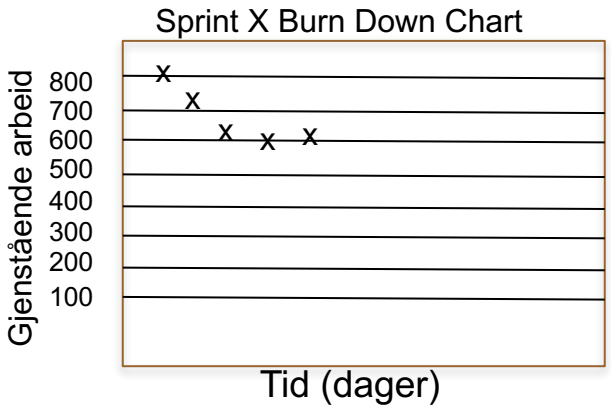
- System spesifikasjonen i XP (og SCRUM) er basert på brukerhistorier (user stories) som reflekterer egenskapene i systemet
  - ”Som student, ønsker jeg å melde meg opp i kurs”
- Teamet diskuterer historier og rangerer dem i forhold til tiden de tror det tar
- Historier som skal være med i en ”iterajon” velges, der antall historier reflekterer tiden det tar å levere en iterajon (typisk 1-4 uker)

Les mer om user stories og forskjellen til use cases (sentralt i UML) på http://en.wikipedia.org/wiki/User_story

SCRUM - prosess og roller

Daily Standup – 3 spørsmål

- Hva har du gjort siden i går?
- Hva planegger du å gjøre til i morgen?
- Hvilke eventuelle hindringer har du?



Scrum Master

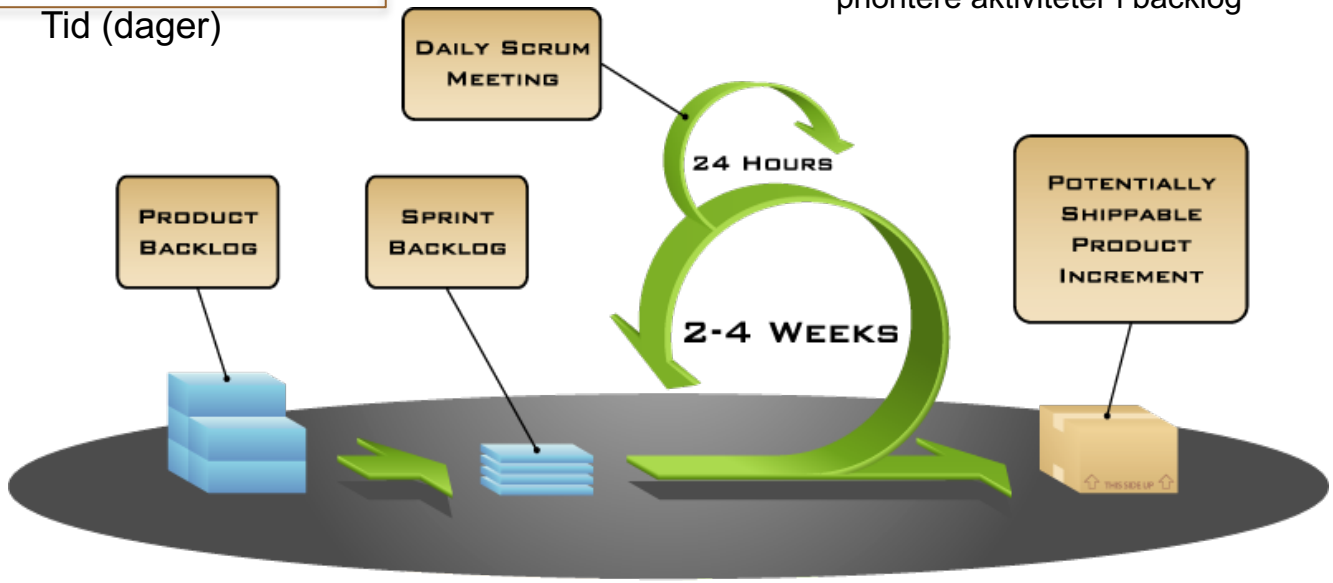
- Sørger for at Scrum prosessen blir fulgt
- Leder "daily stand up"
- Hindrer støy slik at utviklerne kan fokusere på oppgavene
- Teamets "coach og beste venn"
- Koordinerer SPRINT planlegging, og estimering av Sprint backlog

Scrum Team

- Typisk 5-9 personer
- Utviklere, testere, arkitekt,…
  - Jobber fulltid
- Selvstyrt
- Utvikler systemet

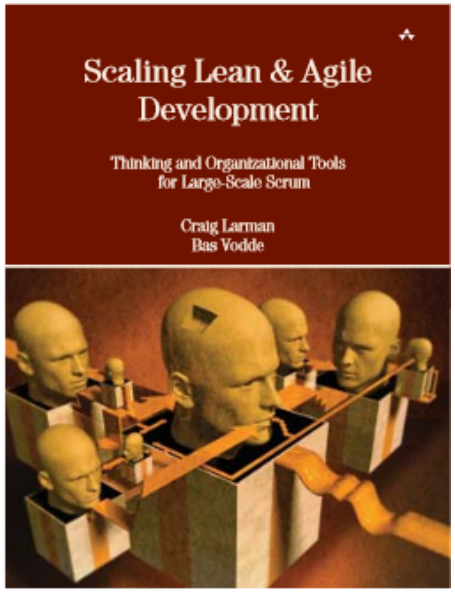
Product Owner

- Representerer kunden
- Setter opp mål for hver SPRINT
- Ansvar for product backlog, og for å prioritere aktiviteter i backlog



Product Backlog (produktkø)

			New Estimates at Sprint ...						
Priority	Item	Details (wiki URL)	Initial Size Estimate	1	2	3	4	5	6
1	As a buyer, I want to place a book in a shopping cart (see UI sketches on wiki page)	...	5						
2	As a buyer, I want to remove a book in a shopping cart	...	2						
3	Improve transaction processing performance (see target performance metrics on wiki)	...	13						
4	Investigate solutions for speeding up credit card validation (see target performance metrics on wiki)	...	20						
5	Upgrade all servers to Apache 2.2.3	...	13						
6	Diagnose and fix the order processing script errors (bugzilla ID 14823)	...	3						
7	As a shopper, I want to create and save a wish list	...	40						
8	As a shopper, I want to to add or delete items on my wish list	...	20						



Product Backlog (produktkø)

- Ulike “items”
  - Kunde spesifikke (Alle kunder skal kunne legge bøker i handlekurven)
  - Forbedringsmål (Skriv om fra C++ to Java)

- “Items” i produktkøen kan utrykkes som
  - User stories
  - Use Cases
  - Andre måter å spesifisere krav på som passer

Sprint planlegging

En måte å lage sprint backlog

				New Estimates of Effort Remaining at end of Day...					
Product Backlog Item	**Sprint Task**	**Volunteer**	**Initial Estimate of Effort**	**1**	**2**	**3**	**4**	**5**	**6**
As a buyer, I want to place a book in a shopping cart	modify database		5						
	create webpage (UI)		8						
	create webpage (Javascript logic)		13						
	write automated acceptance tests		13						
	update buyer help webpage		3						
	. . .								
Improve transaction processing performance	merge DCP code and complete layer-level tests		5						
	complete machine order for pRank		8						
	change DCP and reader to use pRank http API		13						

A Manifesto for Agile Software Developers

The Agile Software Development Manifesto© is an intentionally streamlined expression of the core values of agile project management. Use this manifesto as a guide to implement agile methodologies in your projects.

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work, we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more."

Hvordan manifesto har blitt tolket og ofte praktisert feil

We are uncovering ~~better~~ **the only** way~~s~~ of developing software by doing it and ~~helping~~ **teaching** others do it. Through this work we have come to value:

Individuals and interactions ~~over~~ **and not** processes and tools
Working software ~~over~~ **and not** comprehensive documentation
Customer collaboration ~~over~~ **and not** contract negotiation
Responding to change ~~over~~ **and not** following a plan

That is, ~~while~~ **since** there is **no** value in the items on the right, we value **only** the items on the left ~~more~~.

[Janes & Succi 2012]

Lean –viktige prinsipper

- JIT (Just in Time)
- Tilfredstille kunden
- Flyt
- Visualisering
- Unngå “Waste”
- Støtte opp om endringer

Lean prinsipp: Just-In-Time (JIT)

- Ikke lag noe før det er etterspurt.
- Ikke ta endelige avgjørelser for tidlig.
  - Mange detaljer kan utsettes etter at fundamentale aspekter er implementert.
  - Fokuser kun på detaljer som er nødvendig for å implementere oppgavene som er i prosess.
- Unngå waste:
  - Kan vise seg at enkelte oppgaver ikke er nødvendige å implementere i det hele tatt.
  - Ved å utsette oppgaver blir de ofte bedre implementert enn om de blir tatt tidlig.
  - Reduser flaskehalser.

Smidig prinsipp 1

Vår høyeste prioritet er å
tilfredsstille kunden
gjennom tidlige og kontinuerlige
leveranser
av programvare som har verdi.

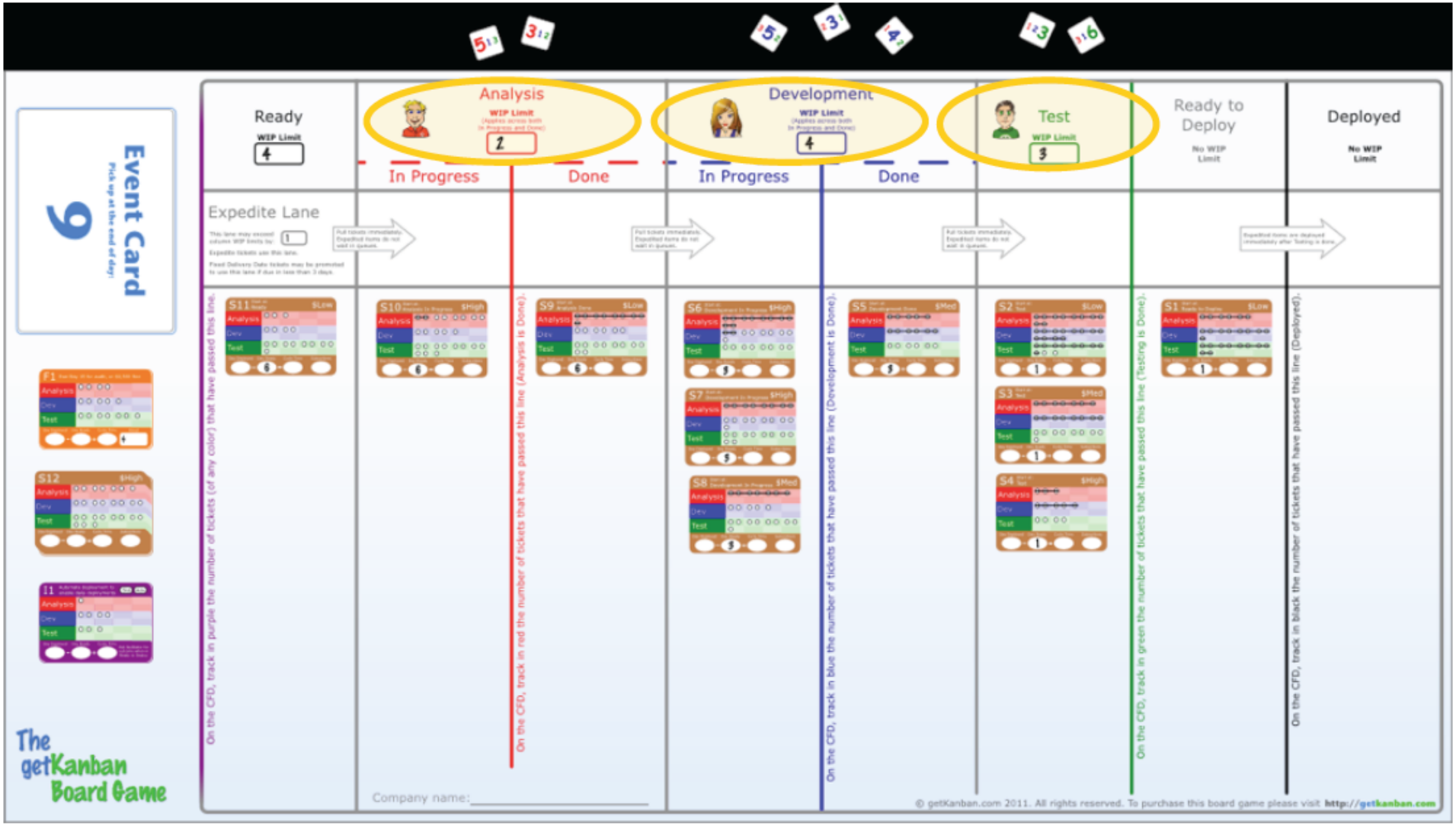
Hvem er kunden?

En som

- betaler for systemet,
- bruker systemet og/eller
- får verdi fra systemet
  - se på hele verdikjeden, inkludert kunder av kundene

- Kunden kan være intern eller ekstern

Visualisering



For å få til flyt, reduser ”waste”

Waste

- Alt som krever ressurser
  - tid
  - arbeids innsats
  - utstyr og lager
  - penger
- Og som ikke gir verdi for kunden

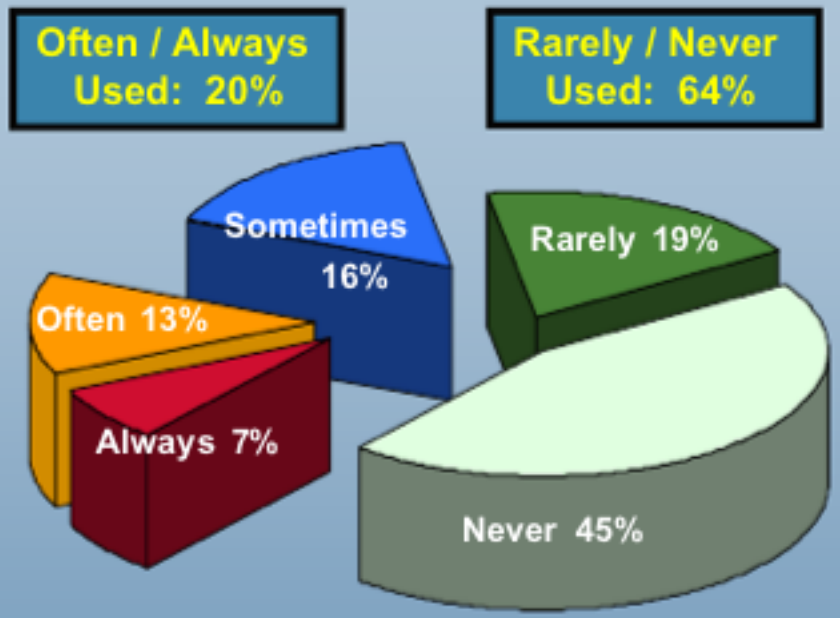
Noen viktige grunner til “waste” i programvareutvikling

- Kompleksitet
- Skalere opp istedet for flyt
- For stort skille mellom de som tar avgjørelser og de som gjør jobben
- Teknisk gjeld

[Poppendieck & Poppendieck 2010]

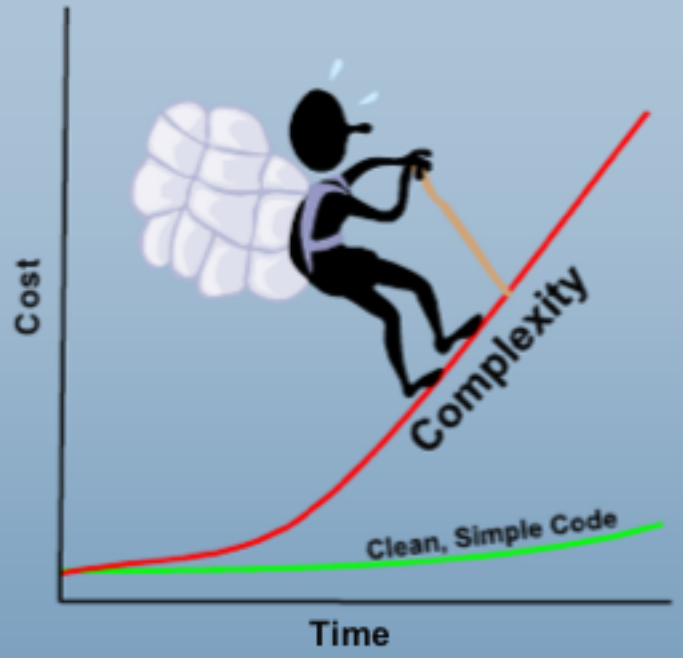
Waste 1: Extra Features

Features / Functions Used in a Typical System



Standish Group Study Reported at XP2002 by Jim Johnson, Chairman

Cost of Complexity



The Biggest opportunity for increasing Software Development Productivity: Write Less Code!

Be a bit sceptical to data from Standish

Grunner til waste: Skalere opp istedet for flyt

- Tradisjonell tanke om at man trenger å skalere opp for å øke profitten
- Effektiv flyt er ofte viktigere
- Utfordringer:
  - Mer fokus på individuelle oppgaver enn på hele verdikjeden
  - For mange oppgaver i paralell (WIP)

Grunner til waste: Skille mellom Business og IT (de som tar avgjørelser og de som gjør jobben)

- Er business og IT sett på som adskilte prosesser, eller er IT betraktet som en del av business prosessen?
- Viktig at ledere skjønner arbeidet, dvs. kan stille relevante spørsmål

Grunner til waste: Teknisk gjeld

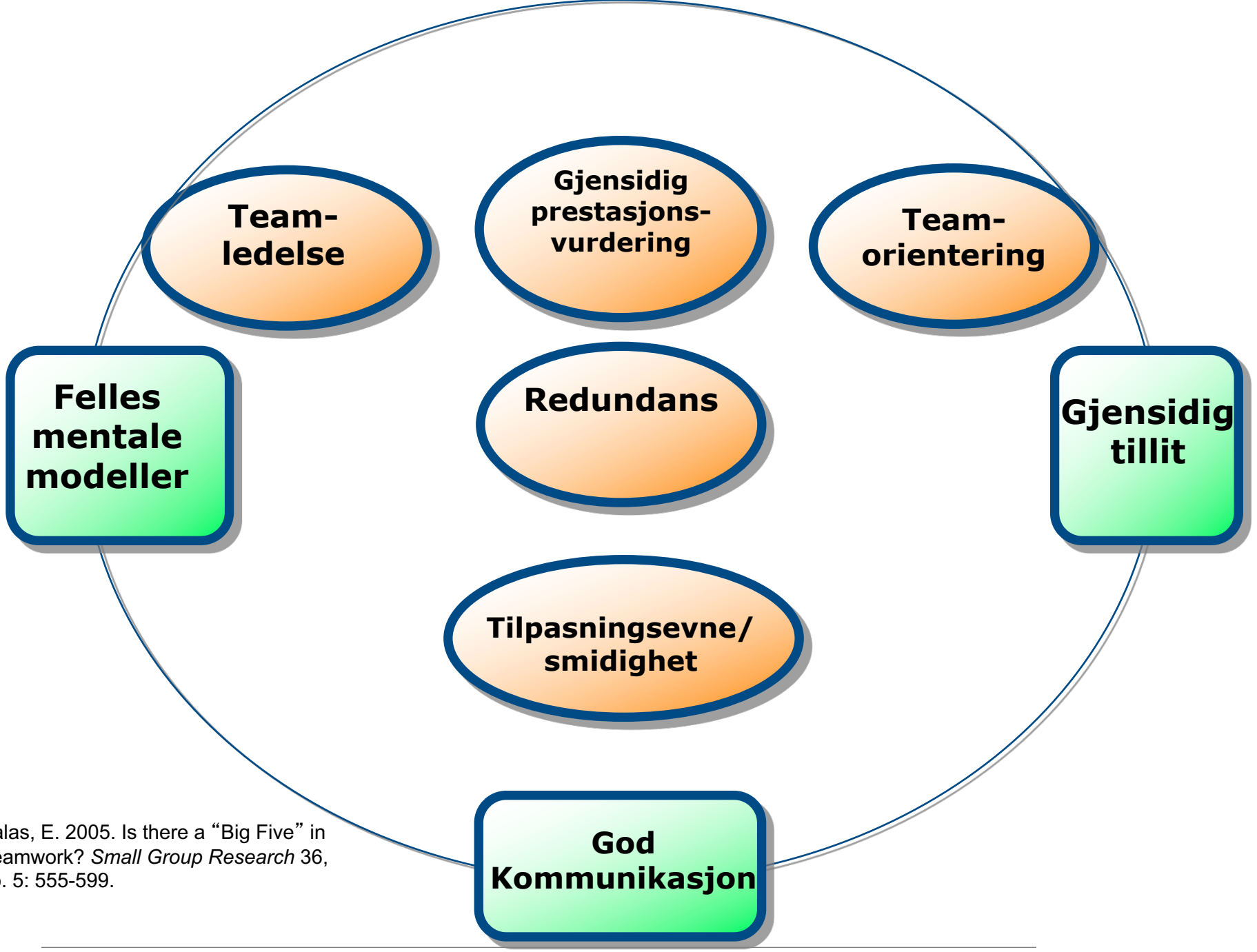
- En metafor som beskriver
  - "shortcuts" som tas i utvikling og vedlikehold for å oppnå kortsiktige fordeler
  - Men som i det lange løp vil føre til økte kostnader på vedlikhold og videre utvikling.

Mer om teknisk gjeld

- = ”Alt som gjør det vanskelig å endre/modifisere kode”
- Siden kode alltid må vedlikeholdes/endres, må vi betale tilbake gjelden.
- Skriv “clean code”, konsis, enkel med få avhengigheter
- Ny funksjonalitet leder til større kodebase og økt kompelsitet; refaktorering reduserer gjelden.

Smidig prinsipp 2

Ønsk endringer i krav velkommen, selv sent i utviklingen.
Smidige prosesser bruker endringer til
å skape konkurransefortrinn for kunden.



Salas, E. 2005. Is there a "Big Five" in Teamwork? *Small Group Research* 36, no. 5: 555-599.

Team effektivitetsmodell

Case – Film og TV-serier

- Appen skal ha støtte for:
  1. Liste populære filmer eller TV-serier
  2. Liste sjangere
  3. Søk i film eller TV-serie - Søkeresultater skal vise tittel og poster / bilde
  4. Liste filmer eller TV-serier innen en sjanger - Liste-elementer skal inneholde tittel og poster / bilde
  5. Detaljert visning av film eller TV-serie - Detaljert skjerm viser stort bilde, tittel, beskrivelse (overview) og utgivelsesår.

Case – Film og TV-serier

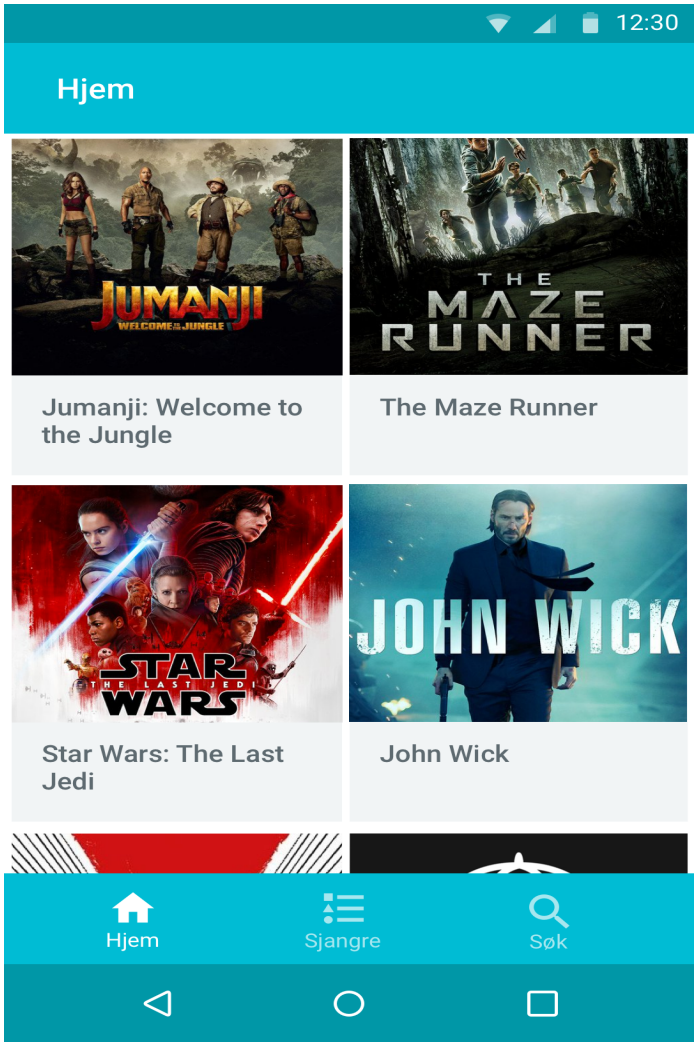
- Appen skal ha støtte for:
  1. Liste populære filmer eller TV-serier
  2. Liste sjangere
  3. Søk i film eller TV-serie - Søkeresultater skal vise tittel og poster / bilde
  4. Liste filmer eller TV-serier innen en sjanger - Liste-elementer skal inneholde tittel og poster / bilde
  5. Detaljert visning av film eller TV-serie - Detaljert skjerm viser stort bilde, tittel, beskrivelse (overview) og utgivelsesår.

Case – Film og TV-serier

- Informasjonen i appen hentes fra APIet til THE MOVIE DB. Se https://developers.themoviedb.org/3 for dokumentasjon. For skjerm 2-5 (se forrige slide) må dere selv finne ut hvilke endepunkter som passer.

- En api nøkkel (API key) er påkrevd, se https://developers.themoviedb.org/3/getting-started/authentication.
- Det er valgfritt om appen skal støtte filmer eller TV-serier.

1. Liste populære filmer eller TV-serier



- Appens ”Hjem” skjerm skal liste opp de 20 mest populære filmene eller TV-seriene akkurat nå.
- Benytt et RecyclerView med liste eller grid (skissen til venstre viser grid).
- Elementer i listen skal vise et bilde (poster) og tittelen på filmen.
- Benytt discover endepunktet med sort_by parameteren. Eksempel på request:
- https://api.themoviedb.org/3/discover/movie?api_key=[API_KEY_HERE]&sort_by=popularity.desc
- Lim inn din egen api key, og benytt Postman for å teste endepunktet.