

INF 5700, 28. september 2012

Ledelse av systemutviklingsprosjekter



UNIVERSITETET
I OSLO

Professor Dag Sjøberg



Plan for forelesningen

- Ledelsesaktiviteter
- Aspekter som påvirker ledelsesaktivitetene
 - System
 - Prosjekt
 - Prosess
 - Kontekst
- Evidens eller systematisert kunnskap som grunnlag for valg av prosesser og teknologier

Prosjektleders ansvar

- Sikre at IT-systemer utvikles, videreutvikles og vedlikeholdes med tilfredsstillende *kvalitet* og innen avtalte *tids-* og *kostnadsrammer*
- Hva må prosjektleder gjøre for å få til dette?

Ledelsesaktiviteter

- Planlegge prosjektet inkl. bestemme og/eller håndtere:
 - kostnader/budsjett
 - tidsplan
 - prosjektmedarbeidere
 - utviklingsteknologi
 - strategi, prosess eller metode for prosjektgjennomføringen
- Lede medarbeidere som er involvert i prosjektet
- Sikre kvalitet fortløpende og håndtere risiko
- Rapportere fremdrift og avvik

Planlegging og forutsigbarhet

- Veldefinerte, repeterbare og planlagte aktiviteter
 - Alle personer vet hva de skal gjøre, hvordan det gjøres (standarder/metoder/verktøy), hva de skal levere og når det skal leveres
- Prosjektplaner og -rapportering
 - Ressursplaner: Kostnadsrammer, personal, utstyr
 - Tidsplaner: Estimering, milepæler, aktivitetsnettverk
- Kvalitetsplaner og -rapportering
 - Sjekklistor, inspeksjoner, testplaner, testresultater ...
 - Rutiner for å håndtere endringsforespørsler, sporbarhet, ...
- Men graden av planlegging og formalitet i systemutvikling er et diskusjonstema
 - Planbaserte versus smidige metoder

Innholdet i ledelsesaktivitetene kommer an på:

- Systemene som skal utvikles
- Kultur, kompetanse og ressurser hos kunden
- Kultur, kompetanse og ressurser i utviklingsorganisasjonen og prosjektet

Typer av systemer

- **Uavhengige systemer:** Kjører lokalt på en datamaskin, f.eks. PC, og trenger ikke koples i noe nettverk
- **Interaktive transaksjonssystemer:** Grensesnittet mot brukerne er på en PC eller terminal, men systemer kjøres på datamaskiner som står andre steder. Eks. Web-applikasjoner
- **Innebygde (embedded) systemer:** programvare som kontrollerer og styrer maskinvare. Eks. mobiltelefoner, togkontrollsystemer
- **Batch-systemer:** Forretningskritiske systemer som prosesserer data i store bolker. Eks. bank og finans
- **Underholdningssystemer:** Spill og annen underholdning
- **Systemer for modellering and simulering:** utviklet av forskere og ingeniører for å modellere fysiske prosesser eller situasjoner
- **Data-innsamlingsystemer:** samler inn data fra omgivelsene ved bruk av sensorer og sender dataene til andre systemer for behandling. Eks. værstasjoner, satellitter

Porteføljer: systemer av systemer

- Slike systemer er ofte “ultra-large scale systems”, dvs. ekstremt komplekse, de endres kontinuerlig, har en rekke interessegrupper og et stort antall maskinvareprodukter, etc.
- Eksempler:
 - World Wide Web
 - Internasjonal flytrafikk-kontroll
 - Porteføljen til store organisasjoner, for eksempel, Telenor og Oslo Universitetssykehus har flere hundre IT-systemer i sin portefølje



Hva er en prosess?

- En prosess er et sett av aktiviteter for å utføre en jobb
- Ønsker normalt å gjennomføre en prosess som er planlagt og gjennomtenkt
- Alt vi gjør involverer prosesser, for eksempel:
 - Planlegge et prosjekt
 - Forberede et møte
 - Utarbeide en kravspesifikasjon
 - Designe en web-side
 - Programmere en funksjon
 - Utføre en test
 - Skrive et dokument
 - ...

Systemutviklingsprosess

- Systemutviklingsprosess (= programvare-prosess) er de aktivitetene som utføres for å utvikle eller videreutvikle et datasystem
- Aktivitetene varierer, men vil alltid ha elementer av
 - spesifisering av kravene, dvs. hva systemet skal gjøre
 - design av systemet (for eksempel lage en datamodell)
 - implementering av koden (programmering)
 - validering av at systemet gjør det kunden ønsker
 - endringer av systemet i forhold til nye og endrede krav hos kunden

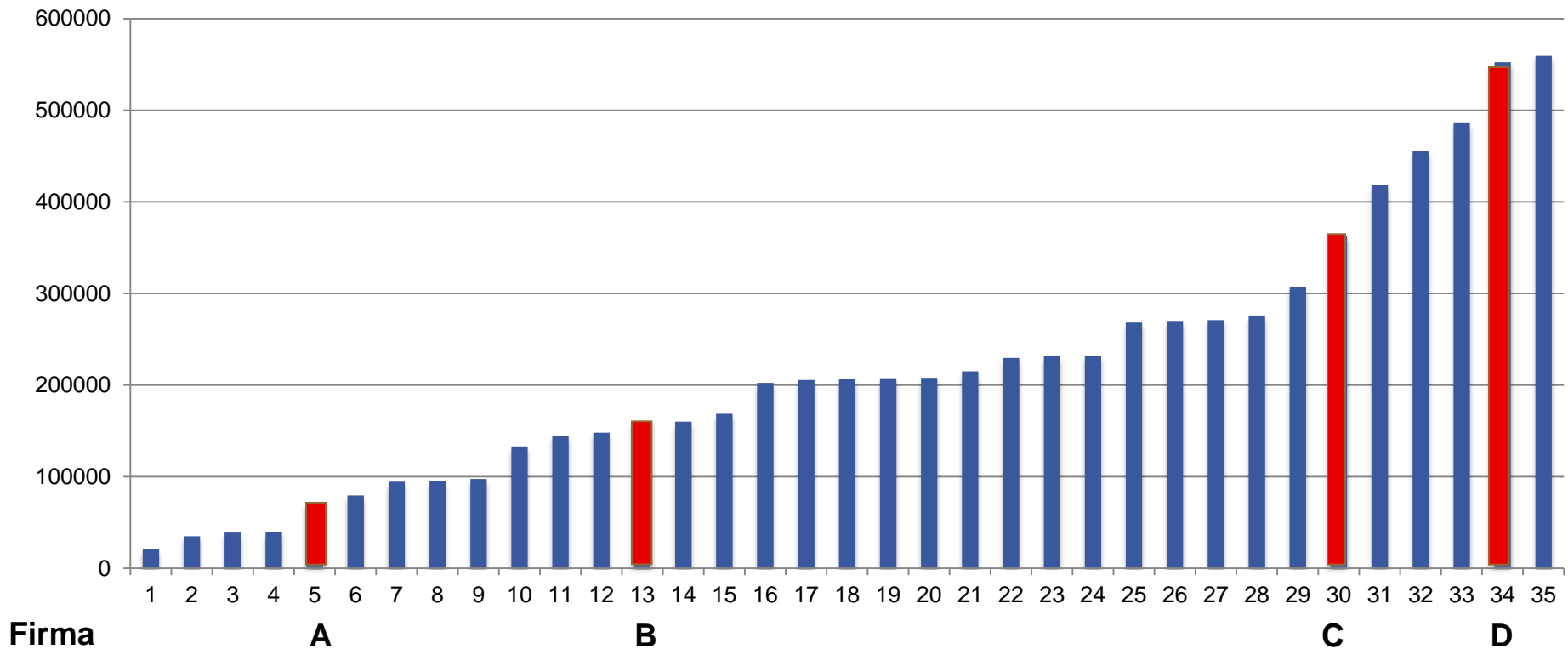
Prosess-egenskaper

- Hvilke aktiviteter inngår i prosessen?
- Hvor mye av hver aktivitet (absolutt og relativt i forhold til hverandre)?
- Når i utviklingsfasen gjøres (hvor mye) av hver aktivitet?
- Prosessbeskrivelser inneholder også
 - delprodukter/resultater (modeller, figurer, tekst, kode etc.) av en aktivitet
 - før- og etterbetingelser (pre- and post-conditions), dvs. betingelser som er sanne før og etter en fase eller et delprodukt er levert
 - rollene til dem som er involvert i prosessen
 - Hvilke og hvordan ulike metoder, praksiser, verktøy og teknikker brukes

Eksempel på ulike utviklingsprosesser

Anbud fra 35 norske og multinasjonale IT-firmaer på et velspesifisert lite, web-basert informasjonssystem (fra kr. 21 000 til kr. 560 000).*
Fire firmaer utviklet det samme systemet uavhengig av hverandre

Tilbud (kroner)



*B.C.D. Anda, D.I.K. Sjøberg and A. Mockus. Variability and Reproducibility in Software Engineering: A Study of four Companies that Developed the same System, *IEEE Transactions on Software Engineering* 35(3):407-429, 2009

Tidsbruk underveis hos de fire firmaene – reflekterer ulike prosesser

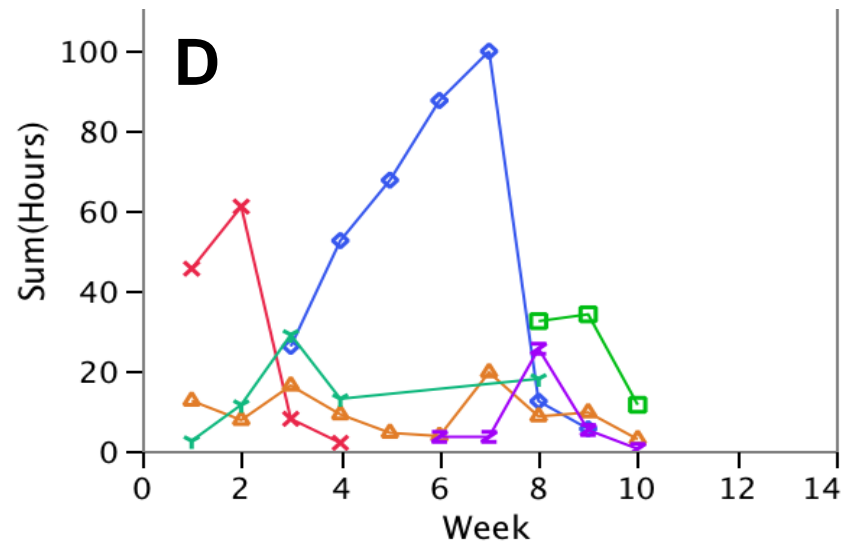
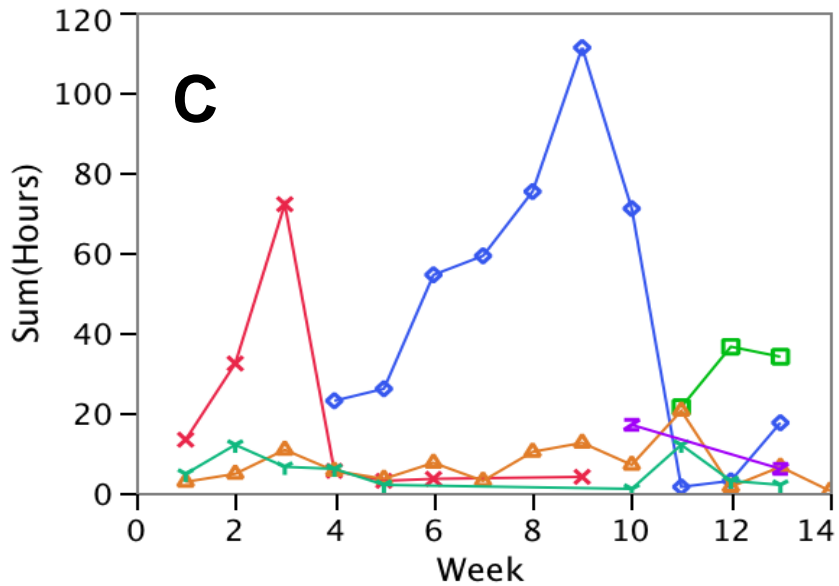
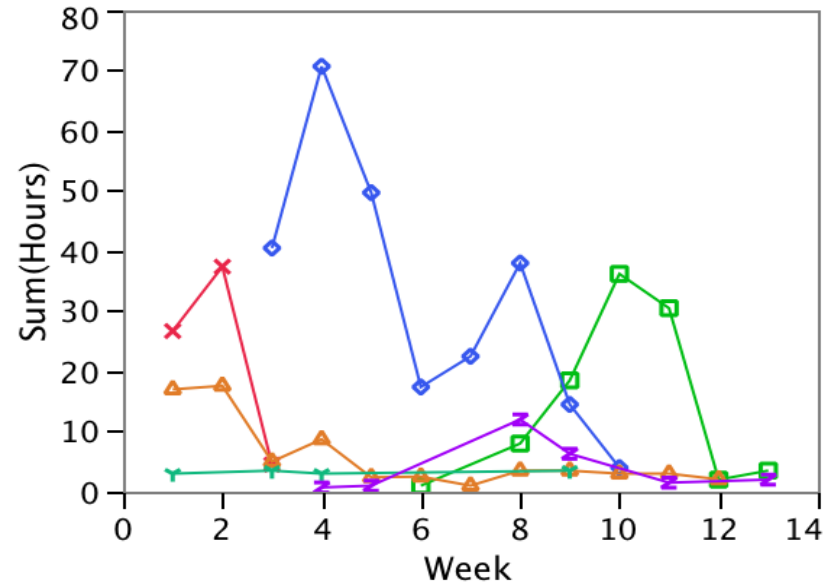
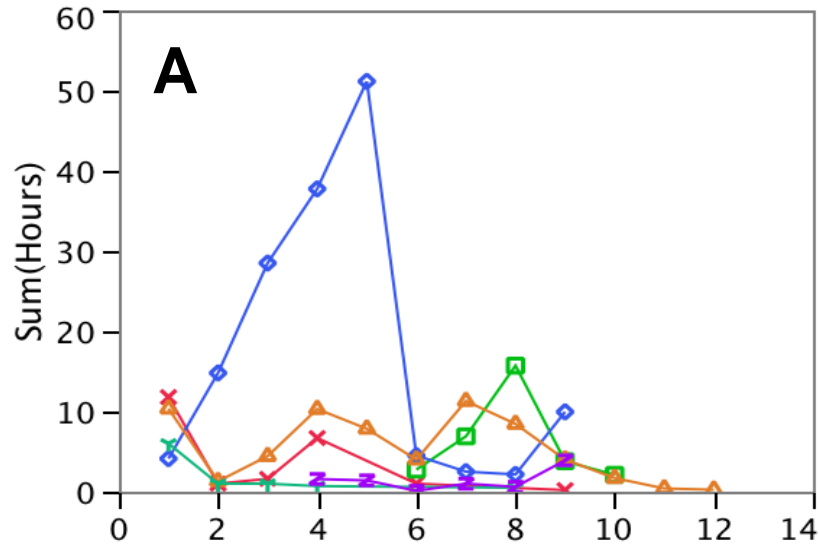
Company=D

0–

Vektlegging underveis

Groups

- × — Activity=Analysis and Design
- △ — Activity=Project Management
- — Activity=Error Correction
- ◇ — Activity=Implementation
- ◇ — Activity=Implementation
- ▽ — Activity=Technical Environment
- ◇ — Activity=Test



Kontekstvariable

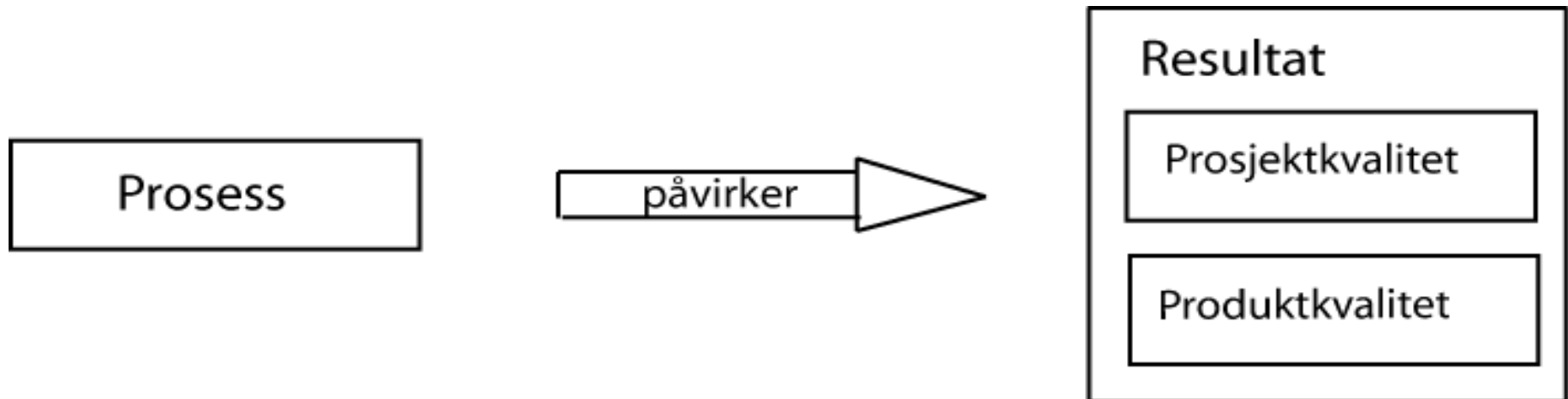
Unit	Variable	Company A	Company B	Company C	Company D
Company	Nationality	Domestic	Domestic	Domestic	International
	Ownership	By employees	Private	By employees	Listed on exchanges
	Location	Bergen	Oslo	Oslo	Oslo + 20 countries
	Size (# employees)	Appr. 8	Appr. 100	Appr. 25	Appr. 13,000 worldwide
	Formal process	Light	Intermediate	Intermediate	Heavy
Project	Firm price	€8,750	€20,000	€45,380	€56,000
	Agreed time schedule	41 days	55 days	73 days	62 days
	Estimated effort	100 hours	220 hours	341 hours	650 hours
	Emphasis on project management	Low	Low	Low	High
	Planned effort on A&D	7%	28%	20%	23%
Team	Allocation	Part-time	Part-time	Part-time	Full-time
	Co-location	No	No	No	Yes
	Turn-over	No	Change of developer	No	No

Prosessvariable – eksempler

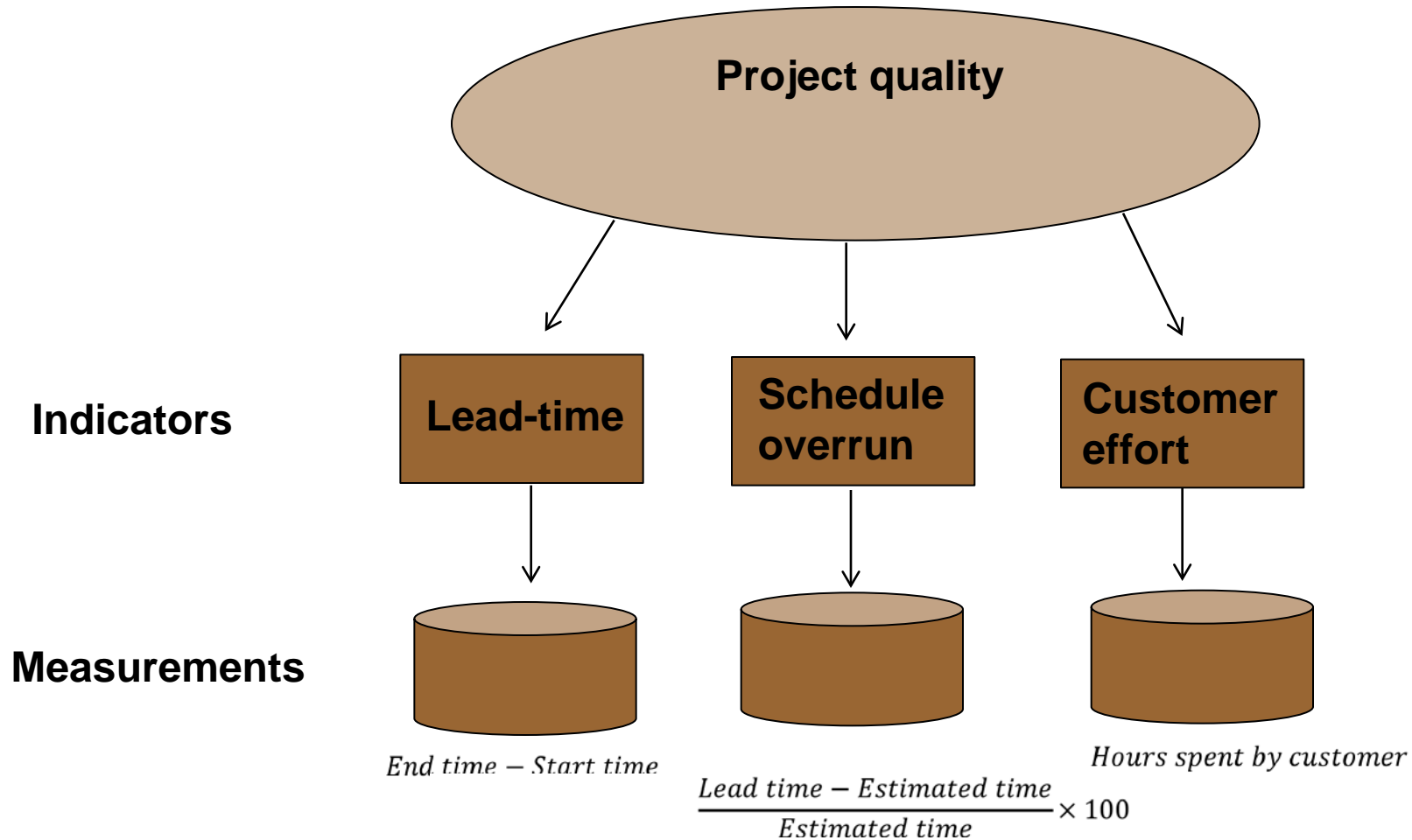
Dimension	Variable	Company			
		A	B	C	D
Work hours	Regular hours	No	Yes	No	Yes
CM Tool	Commits	Small	Large	Large	Small
	Login	Same	Different	Different	Same
Language	JSP usage	High	Low	Low	Low
Issues with customer before acceptance test	Project management	Low	High	Medium	Low
	Functional clarifications	Low	Medium	High	Medium
	Graphical design	Low	Medium	Low	High
	Technical issues	Medium	Medium	Medium	Medium
	<i>Overall</i>	<i>Low</i>	<i>High</i>	<i>High</i>	<i>High</i>
Rework	Deleted/Added	Low	High	High	Medium
	Bugs in acceptance test	Many	Medium	Medium	Few
Emphasis on Activity and Phase	Analysis & Design	Low	High	High	High
	Error correction	Medium	High	Medium	Medium
	Test	Low	Medium	Medium	High
	Tail heavy	No	No	Yes	No

Proessen påvirker resultatet

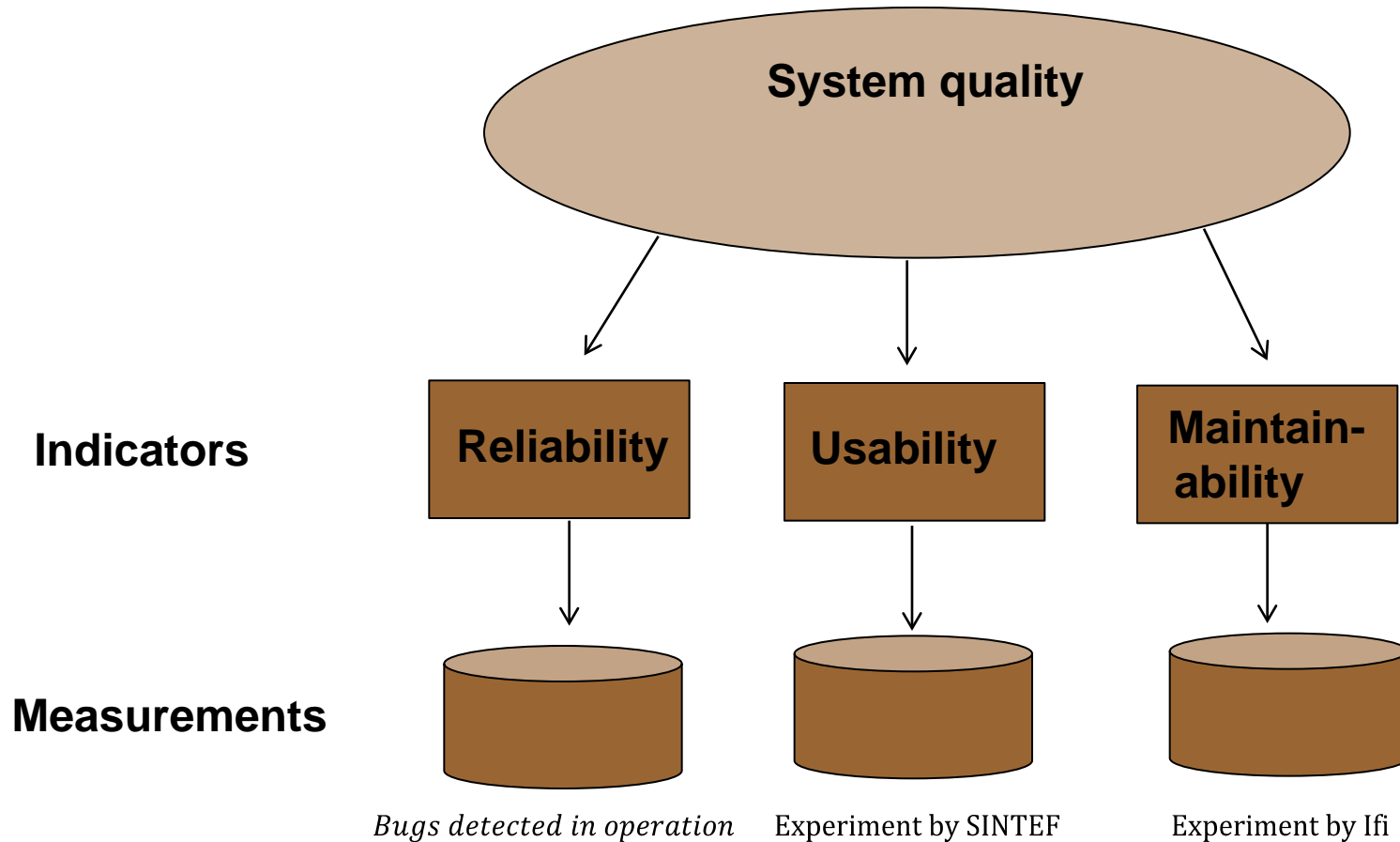
- Systemutviklingsprosessen, dvs. måten man jobber på, i et utviklingsprosjekt vil påvirke kvaliteten både på prosjektet selv og systemet som utvikles
- Måten man jobber på påvirker også arbeidsmiljøet (trivsel, motivasjon, kompetanseutvikling etc.) som igjen påvirker prosjekt- og produktkvalitet generelt



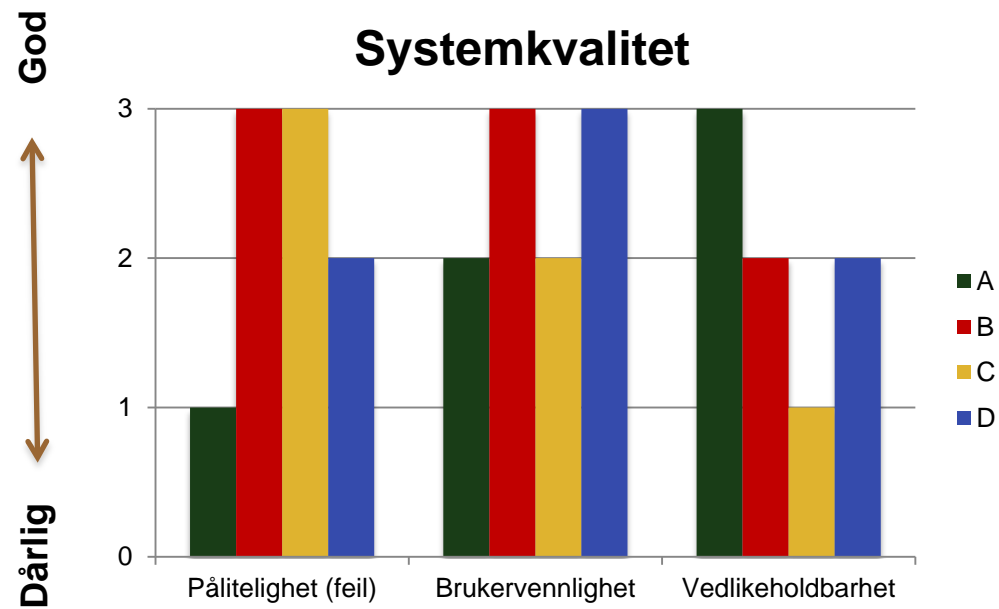
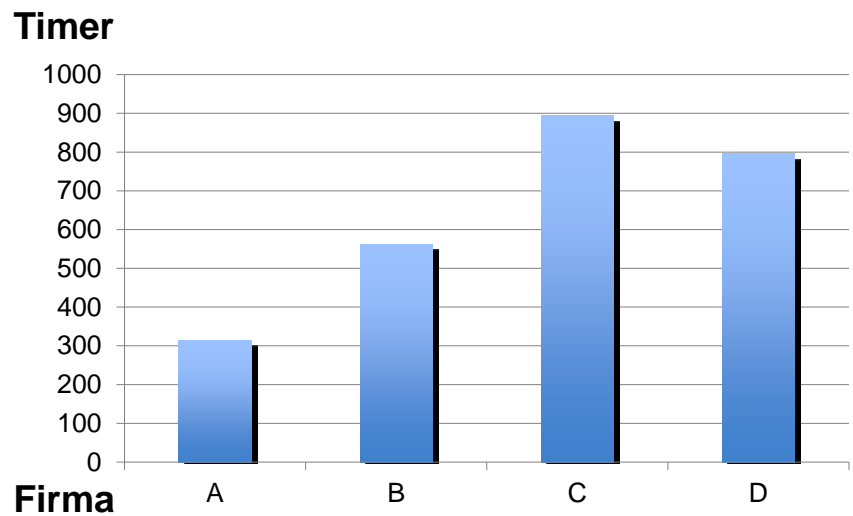
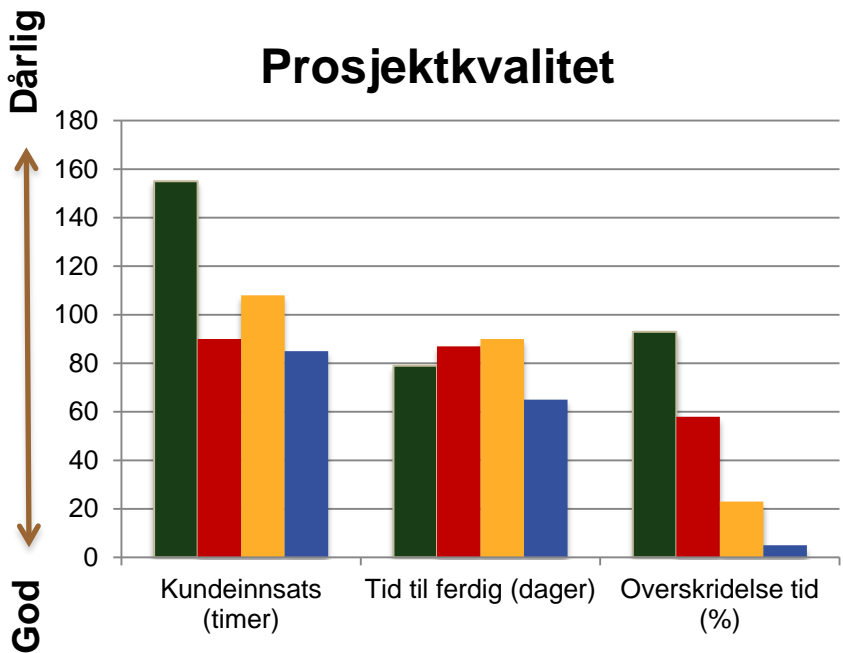
Measuring a Project Quality (this case)



Measuring a System Quality (this case)



Resultat



Del-oppsummering

- Det finnes ulike prosjektkvalitetskriterier
- Det finnes ulike systemkvalitetskriterier
- Hva utviklerorganisasjonen eller kunden vil prioritere, påvirker hvordan man leder et programvare-prosjekt

Valg av prosess og prosessmodell

Anta at du er IT-leder (prosjektleder, avdelingsleder, IT-sjef etc.), hvilke prosesser ønsker jeg i min avdeling eller i mine prosjekter?

Modeller for utviklingsprosesser

Utviklingsstrategi	Definer alle Krav først?	Flere utv.-sykluser?	Distribuer inkremitter?
Fossefall	Ja	Nei	Nei
Inkrementell	Ja	Ja	Kanskje
Evolusjonær	Nei	Ja	Ja

Waterfall model: A linear sequence of steps: krav (black), design (purple), kode (blue), test (grey), lever (orange).

Incremental model: A sequence of steps: krav (black), design (purple), kode (blue), test (grey), lever (orange), followed by a second sequence: design (purple), kode (blue), test (grey), lever (orange).

Evolutionary model: A circular flow of steps: krav (black), design (purple), kode (blue), test (grey), lever (orange), with arrows indicating feedback loops between adjacent and non-adjacent steps.

Reell prosess versus modell av prosess

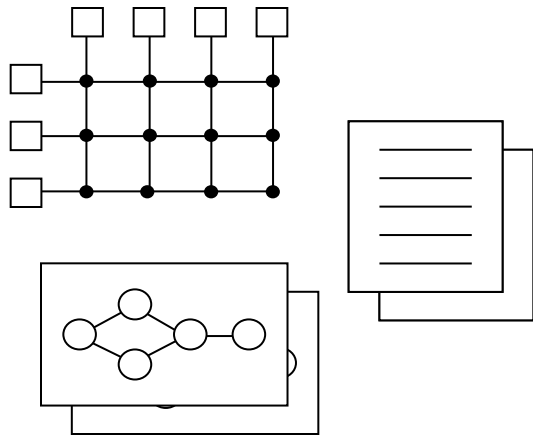
- Systemutviklingsprosess (= faktisk, reell prosess):
 - de aktivitetene som utføres i et utviklingsprosjekt
- Prosesmodell (=formell prosess)
 - En abstrakt representasjon av en prosess
 - Deskriptiv: beskriver en prosess slik vi mener vi utfører den
 - Normativ (preskriptiv): beskriver en prosess slik noen mener den *bør* være (vanligste betydning)

Modell versus virkelighet



Formell versus reell prosess

Det vi sier vi gjør eller
det vi bør gjøre



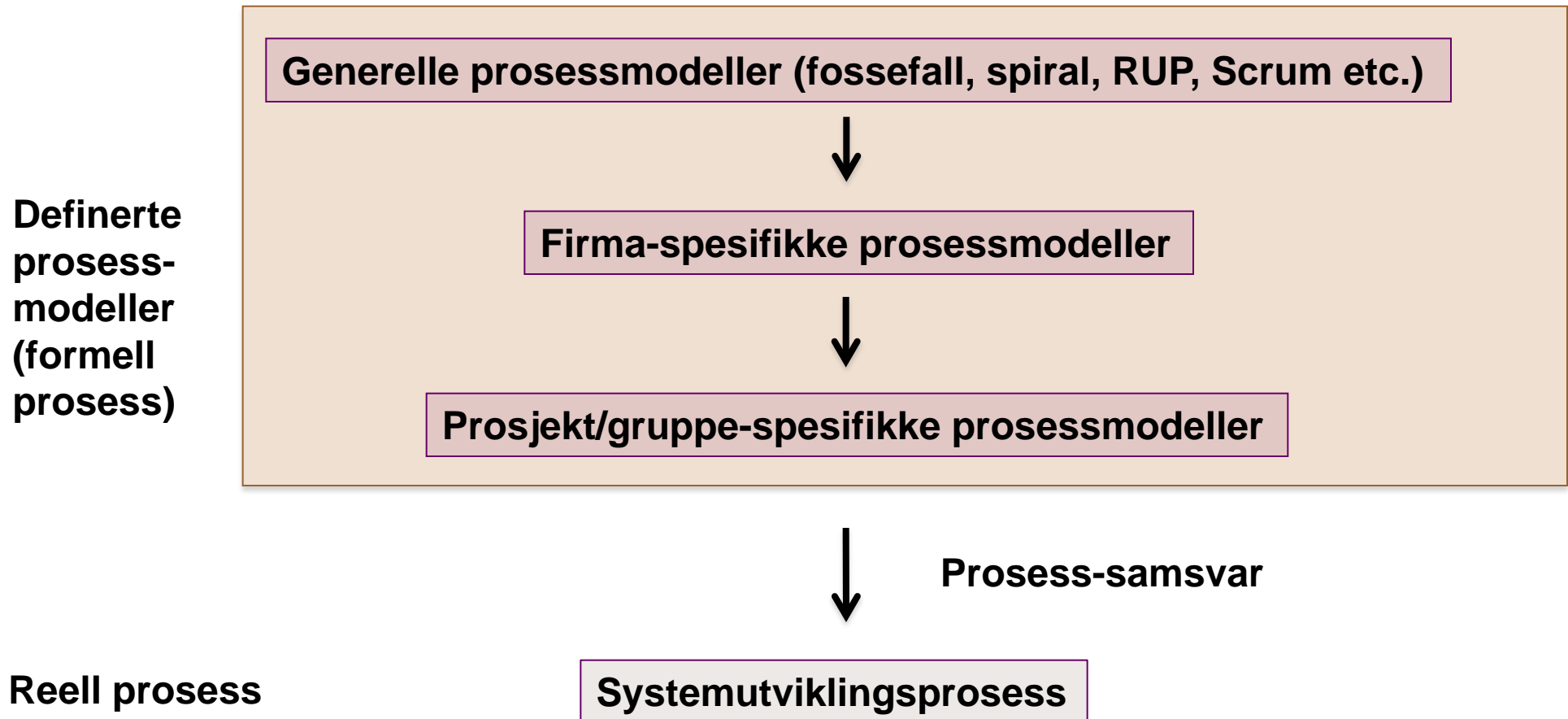
Prosessbeskrivelse

Det vi gjør



Prosessutførelse

Nivåer av prosessmodeller



Hvordan *tilpasse* prosesser?

- Prosesser må tilpasses – ingen prosjekter er like
 - Mange faktorer påvirker prosessen
- Hva kan tilpasses?
 - Antall faser/aktiviteter, roller, ansvarsforhold, dokumentformater, formalitet/frekvens på rapporter og gjennomganger
- Hvordan tilpasse?
 1. Identifiser prosjektomgivelser – utviklingsstrategi, risiko, krav
 2. Innhent synspunkter fra utviklere, brukere, kunder
 3. Definer prosesser, aktiviteter og roller
 4. Dokumenter og begrunn tilpasningene

Prosessprinsipper, for eksempel timeboxing versus “task-boxing” / “task-flyt”

Scrum

- Velg noen prioriterte oppgaver og jobb med dem i faste tidsintervaller (“sprinter” på 2-4 uker) med definerte oppstarts- og avslutningsaktiviteter (tidsboksing).
- Ikke alltid greit å dele inn oppgaver eller “features” av systemet tilpasset disse tidsintervallene, f.eks. vedlikehold, videreutvikling og support.

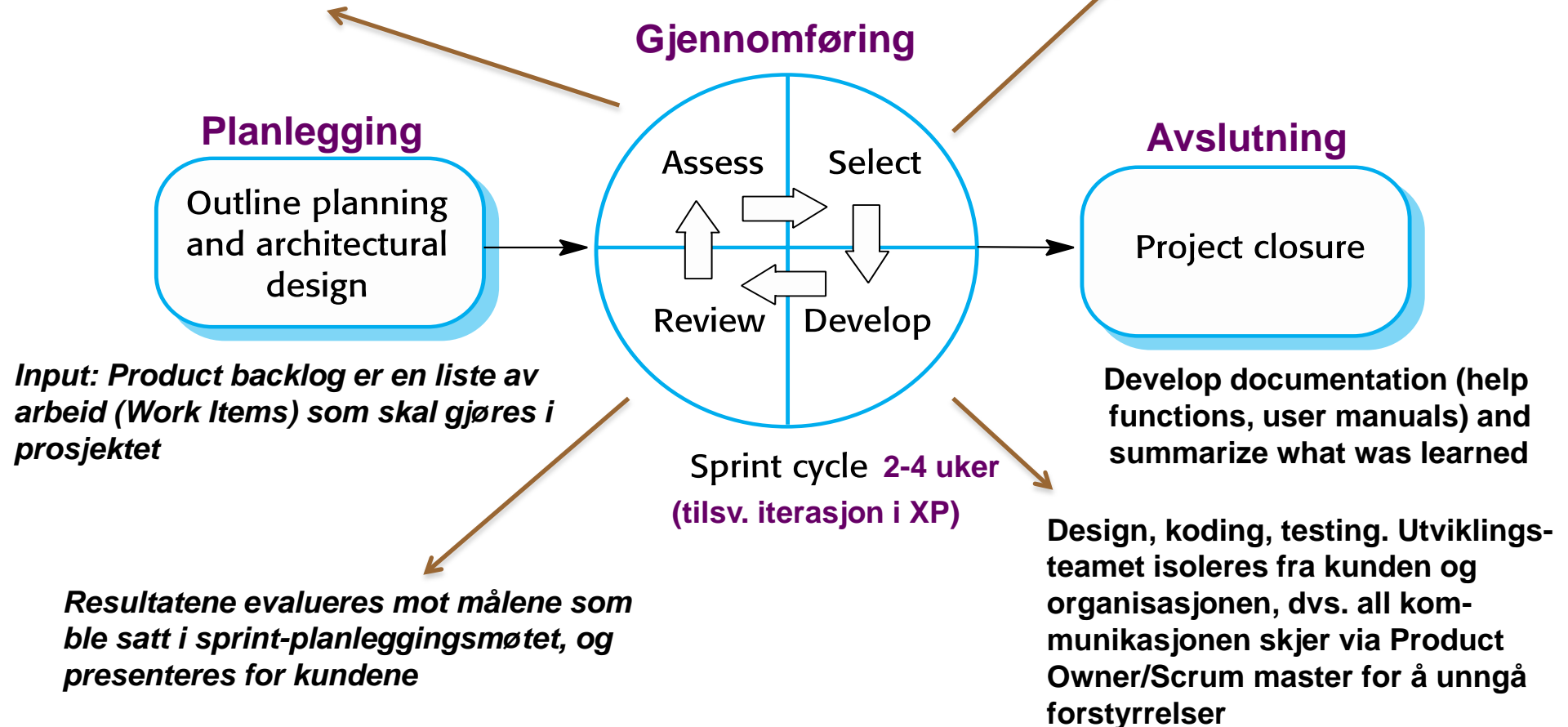
Kanban

- Definerer et sett med oppgaver eller “features” og lever så snart man er ferdig. Oppgaver skal “flyte” uten avbrudd gjennom de nødvendige aktivitetene til de er ferdige (oppgaveboksing)

Scrum

I sprint-planleggingsmøtet evalueres produkt-backloggen som er en samling av brukerhistorier. Mål for sprinten settes inkl. prioriteter og risiko. Kunden kan sette nye krav el. gi nye oppgaver

Utviklingsteam og kunde velger egenskaper og funksjonalitet som skal utvikles i sprinten



Teamarbeid i Scrum

(kan også gjøres med Kanban)

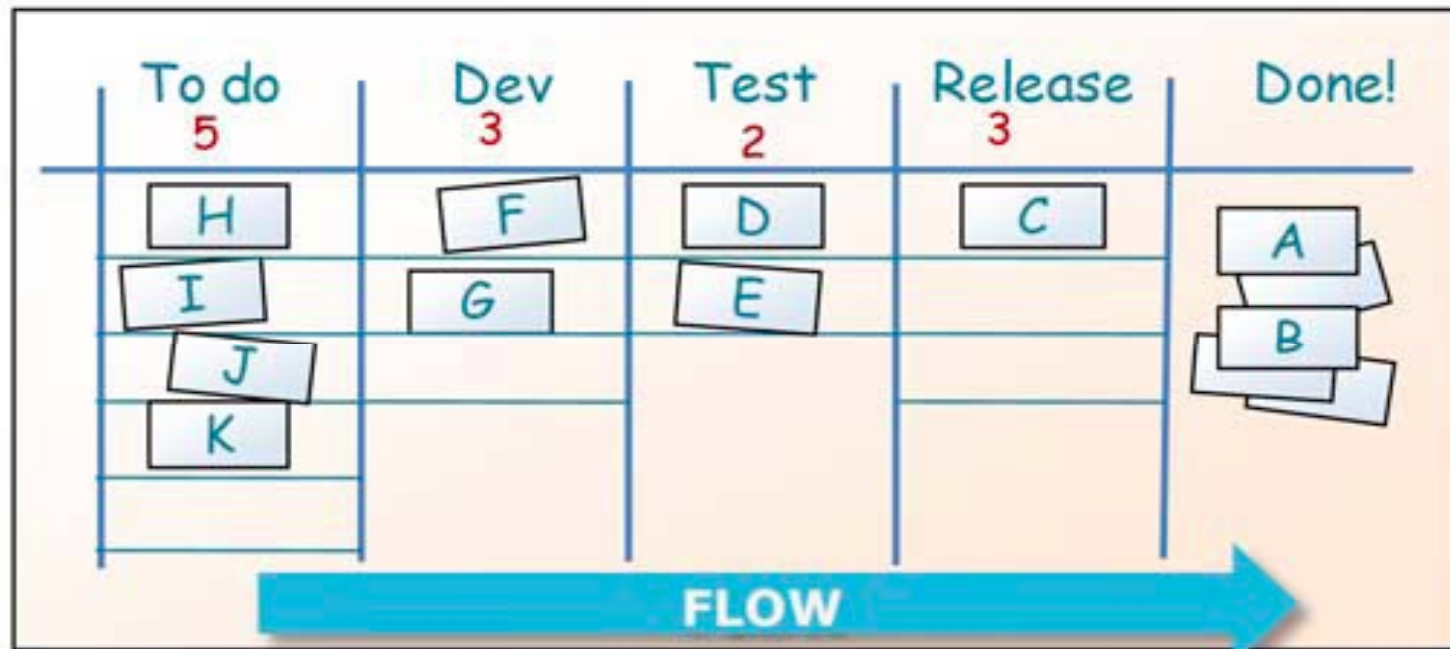
- ‘Scrum master’ er en tilrettelegger som
 - arrangerer daglige møter
 - holder orden på (back)loggen av arbeid som skal gjøres
 - noterer beslutninger
 - måler progresjon i forhold til loggen
 - kommuniserer med kunder og ledelse utenfor teamet
- Hele teamet deltar i korte, daglige møter der alle
 - deler informasjon
 - beskriver progresjon siden forrige møte
 - diskuterer problemer som har oppstått
 - planlegger neste dag
- Varierer hvor godt daglige møter fungerer (dr.grad Viktoria Gulliksen Stray)

Brukerhistorie (user story)

- Én eller flere setninger som beskriver hva brukeren av et system ønsker å få ut av systemet på formen:
 - ”Som en <rolle> ønsker jeg <funksjon> for å oppnå <verdi>”
- Beskrivelsen skal være kort slik at den passer på et kort eller gul lapp

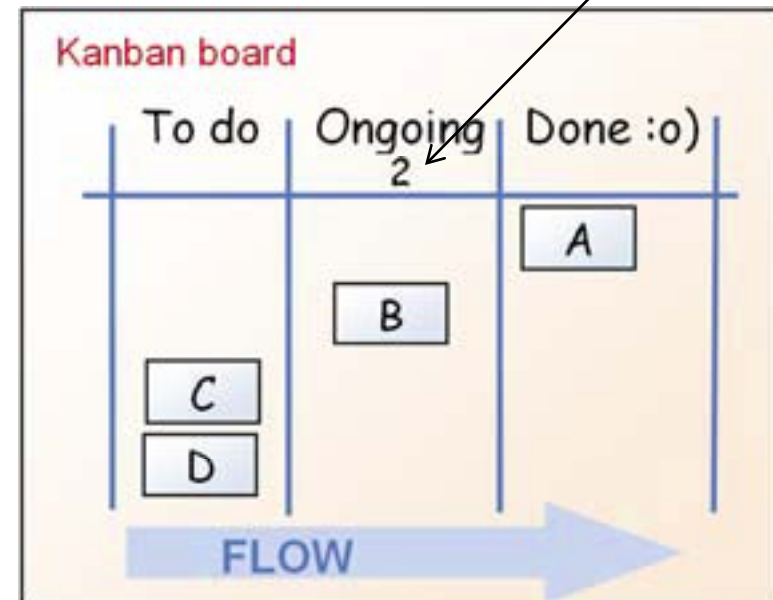
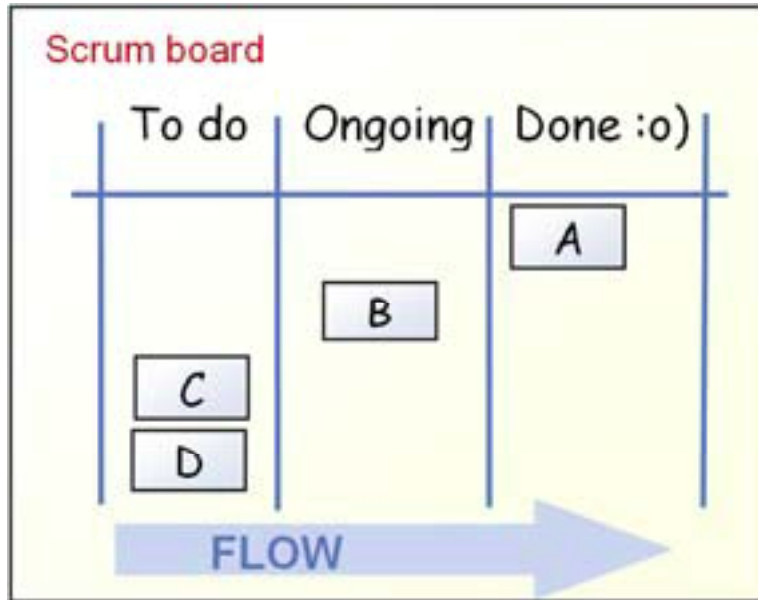
Kanban board

- A Work Item represents a unit of work to be carried out by the development team
- Describe a Work item on a post-it sheet and put it on a board in one of the categories : **"To do"**, **"In progress"** or more detailed states. **"Done"** shows the Work Items that are finished



Scrum board versus Kanban board

Max WIP



From: Kanban and Scrum - making the most of both by Henrik Kniberg and Mattias Skarin on Dec 21, 2009



Bør man bruke Scrum eller Kanban?

Overgang fra Scrum til Kanban i Software Innovation (SI)

- SI begynte med Scrum i 2007 og Kanban i 2010
- Motiver for overgang til Kanban:
 - Økt produksjonen
 - Bedre systemkvalitet, bl.a. ved å redusere lead-time
- Har forventningene blitt møtt?
- Analyse av 12 000 utviklingsoppgaver (work items) over 3,5 år registrert i Team Foundation Server (TFS)

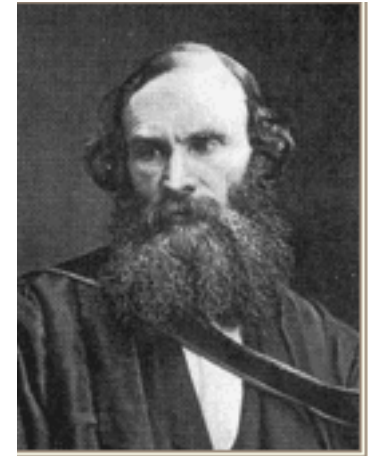
Dag I.K. Sjøberg, Anders Johnsen and Jørgen Solberg: Quantifying the Effect of Using Kanban versus Scrum: A Case Study. *IEEE Software*, Vol. 29, Nr. 5, side 47–53, Sep./Oct. 2012

Sagt om måling

*To measure is to know.
If you cannot measure it, you cannot improve it.*
Lord Kelvin

*Not everything that counts can be measured.
Not everything that can be measured counts.*
Albert Einstein

In God we trust, all others bring data –
W. Edwards Deming



Lord Kelvin

Måling av prosesskvalitet

Eksempel fra [Sjøberg et al. IEEE SW 2012]

	Name	Values
Independent variables	Process	Scrum or Kanban
	Type of work item	Bug or Project Backlog Item (new features, adaptive maintenance tasks, and support tasks, i.e., all tasks that are not bug fixing)
Control variable	Year.Quarter	Each quarter from 2009.1 to 2011.4
	Churn	Number of lines added, deleted, or modified
Dependent variables	Lead time	Number of days from “Next” state to “Ready for release” state on the board
	Production	Number of work items developed per quarter (often called “throughput” [3])
	Productivity	Production per developer
	Productivity 2	Total churn per developer per quarter
	Quality	Number of weighted bugs in the severity levels: Blocking (weight 8), Critical (4), Moderate (2), and Minimal (1)

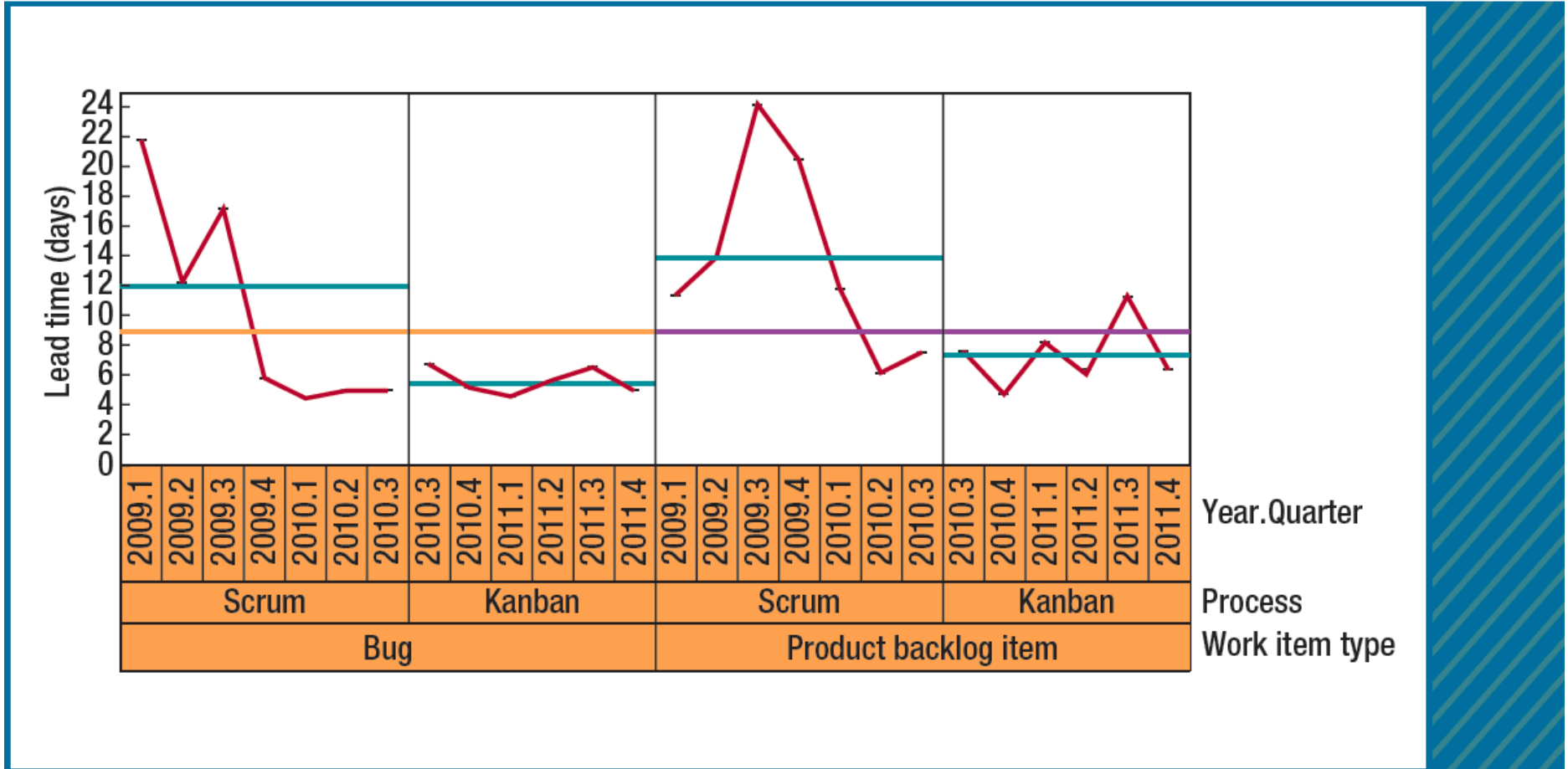


FIGURE 1. Average lead time measured in days by work item type, process, and quarter. The average lead time declined by approximately 50 percent from the Scrum period to the Kanban period. For bugs, the average lead time fell from 12 days for Scrum to five for Kanban. For the project backlog items (PBIs), the lead time declined from 14 to seven days. The orange and purple lines indicate that the bugs and PBIs had the same (weighted) average lead time (nine days) over the whole period. The local top on each third quarter is due to less activity during the summer holiday.

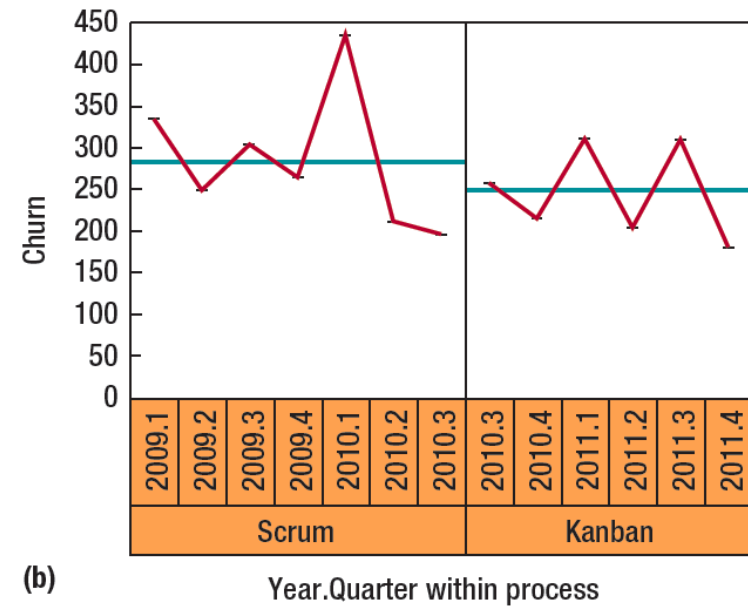
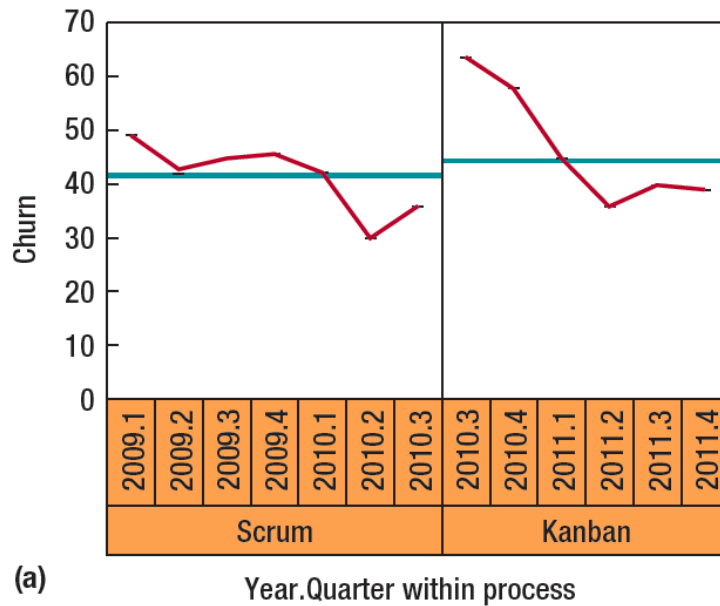


FIGURE 2. Average churn of (a) bugs and (b) PBIs. The average churn for bugs is 6 percent higher in the Kanban period than in the Scrum period, while for PBIs, the average churn is 12 percent lower in the Kanban period than in the Scrum period.

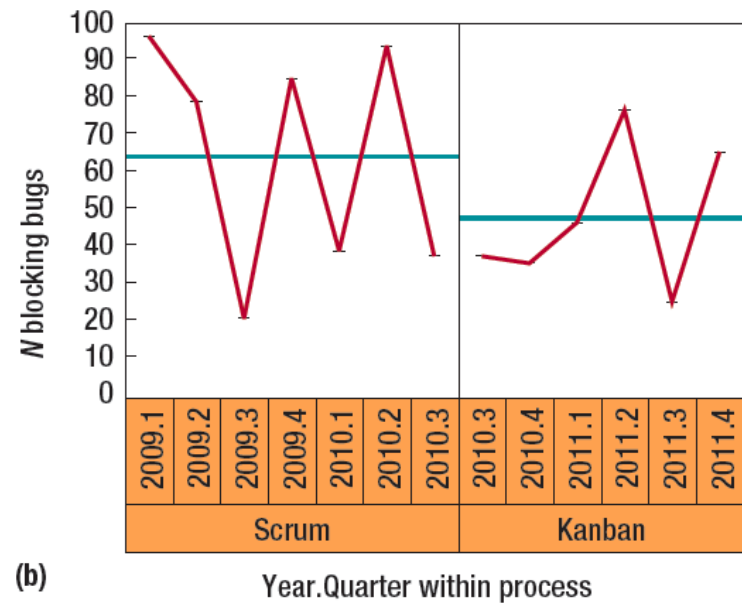
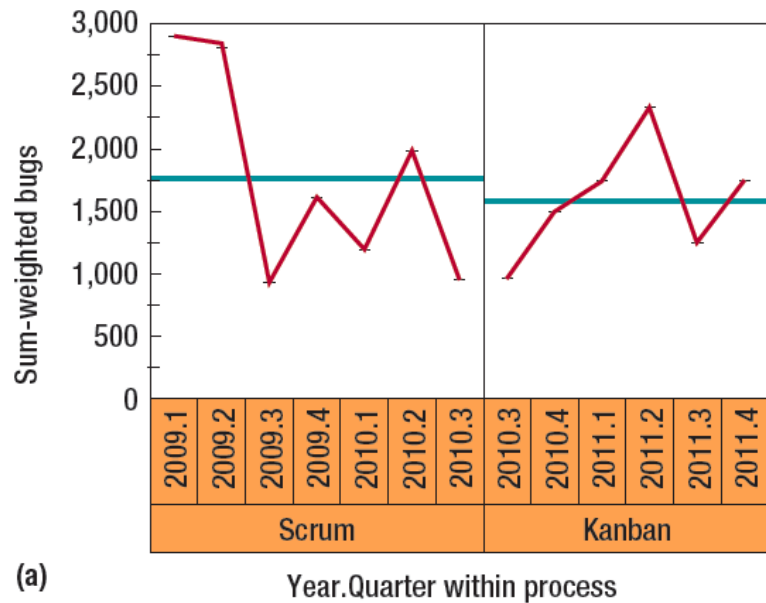


FIGURE 3. Bugs: (a) weighted and (b) blocking. The average number of weighted bugs per quarter fell from 1,774 in the Scrum period to 1,591 in the Kanban period. The most critical bugs, blocking bugs, declined in number even more between the two process periods (from 65 to 48).

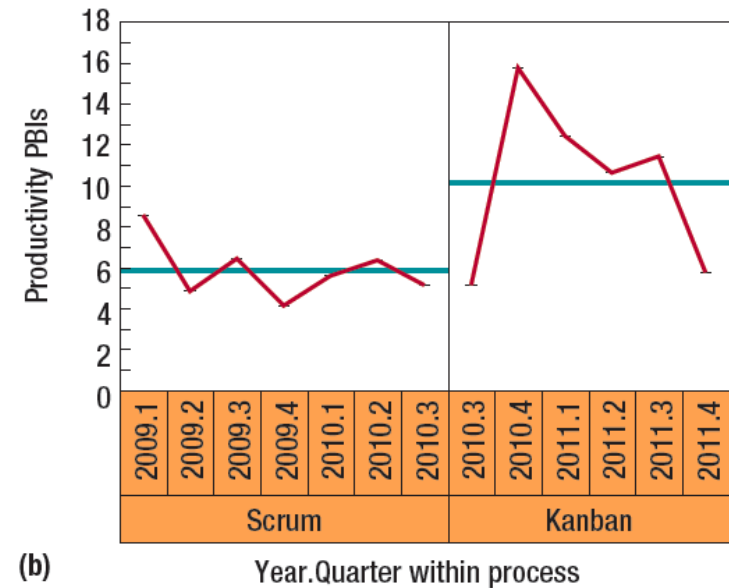
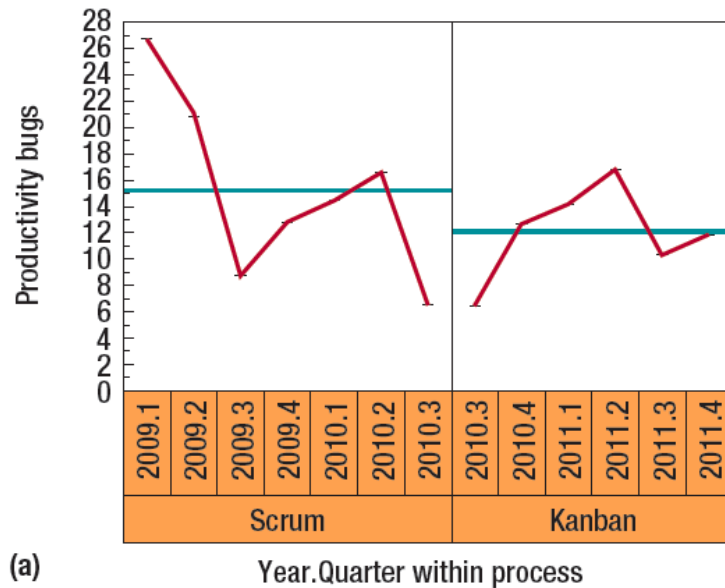


FIGURE 4. Productivity: (a) bugs per developer and (b) PBIs per developer. The number of work items per person, or productivity, decreased from 15.3 to 12.1 for bugs but increased from 5.9 to 10.2 for PBIs.

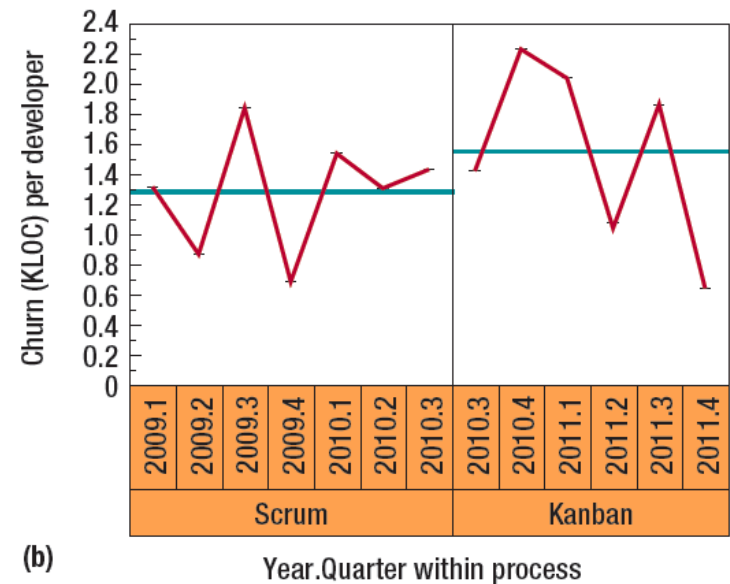
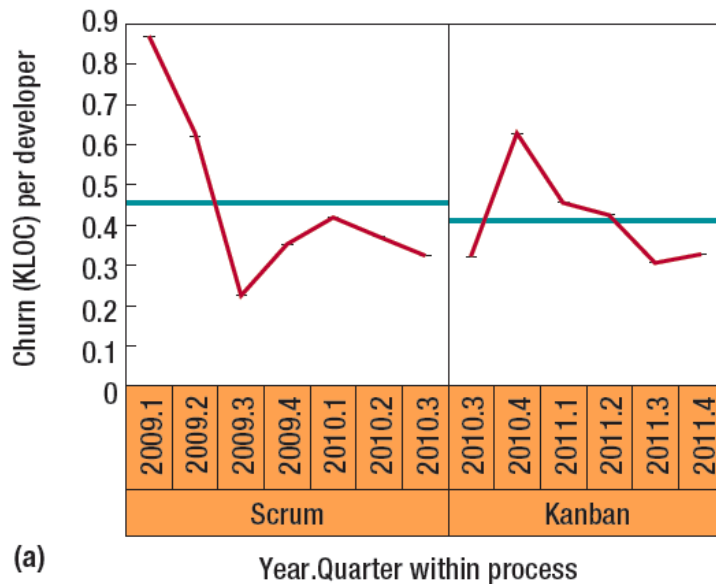


FIGURE 5. Productivity 2: (a) bugs per developer and (b) PBIs per developer. For bugs, productivity decreased from an average of 0.46 KLOC for Scrum to 0.41 KLOC for Kanban; for PBIs, productivity increased from an average of 1.28 KLOC for Scrum to 1.55 KLOC for Kanban.

Konklusjon på studien

- Ved å erstatte Scrum med Kanban, SI
 - nesten halverte ledetiden
 - reduserte antall feil med 10%
 - forbedret produktiviteten
- SI ser derfor ut til å dra fordel av å bruke Kanban istedenfor Scrum
- Bør vurderes av andre som erfarer
 - vanskeligheter med estimering
 - avbrytelser pga. ad hoc-bug fiksing, support og vedlikeholdsoppgaver