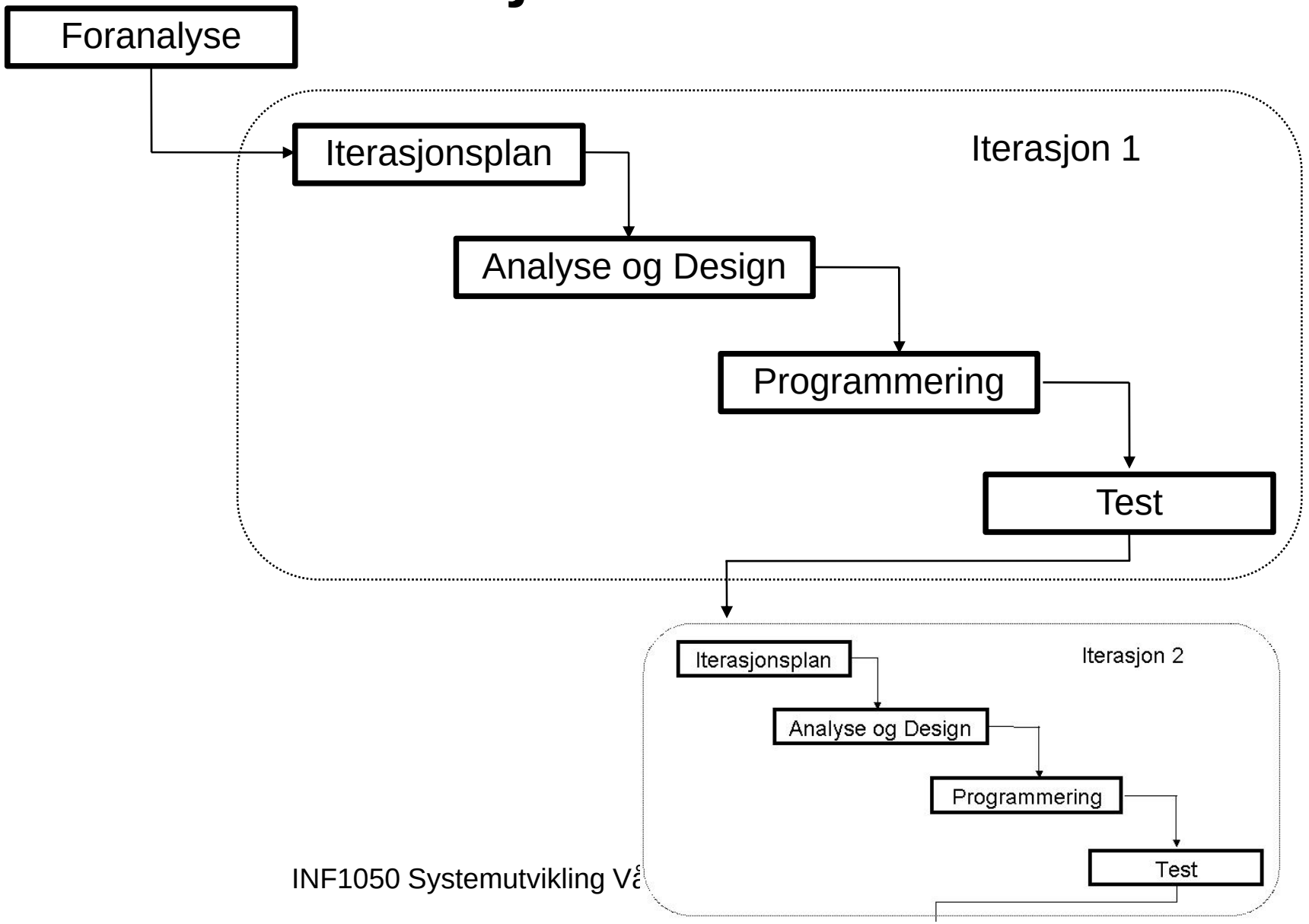
Systemutviklingssprosesser, prosjektarbeid

Forelesning 3 - INF1050 Systemutvikling 28.1.2009

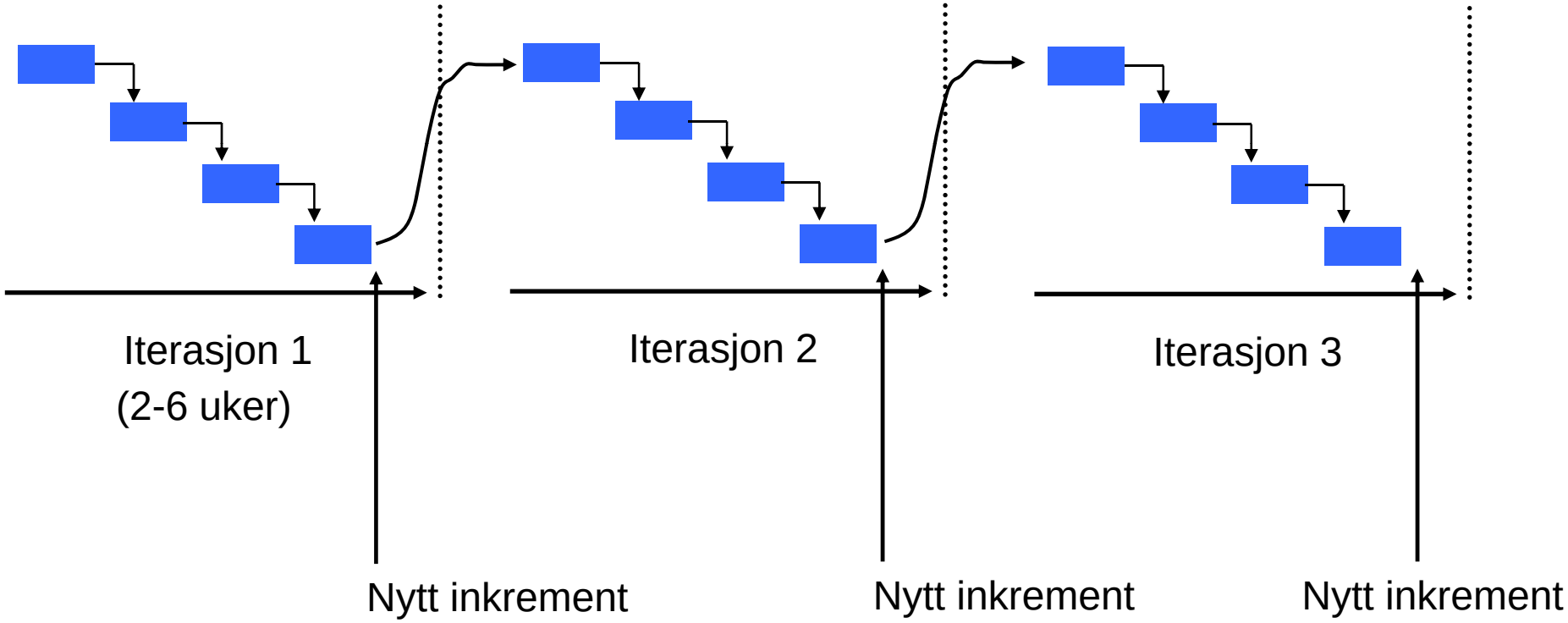
Rune Steinberg

International Development Manager ERP

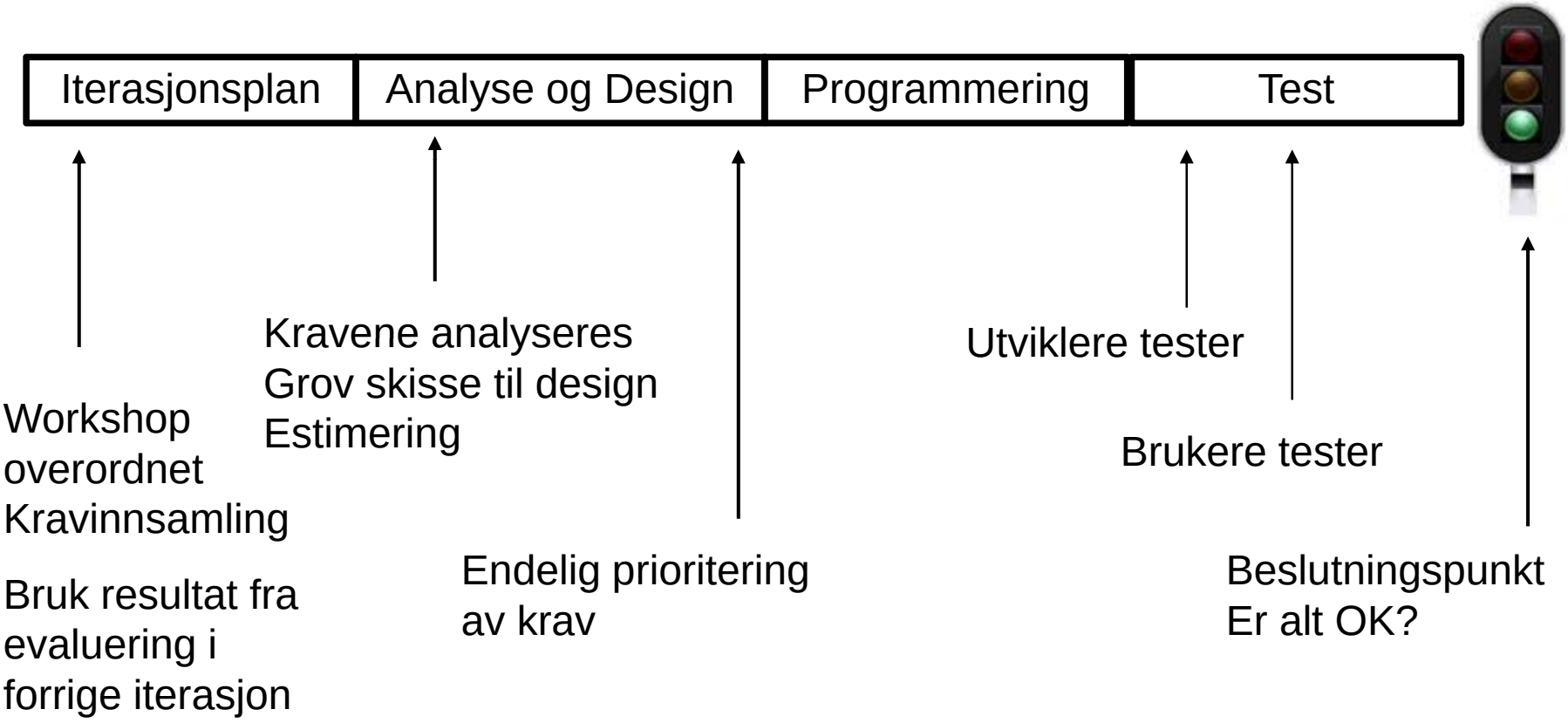
Evolusjonære modeller



Evolusjonære modeller



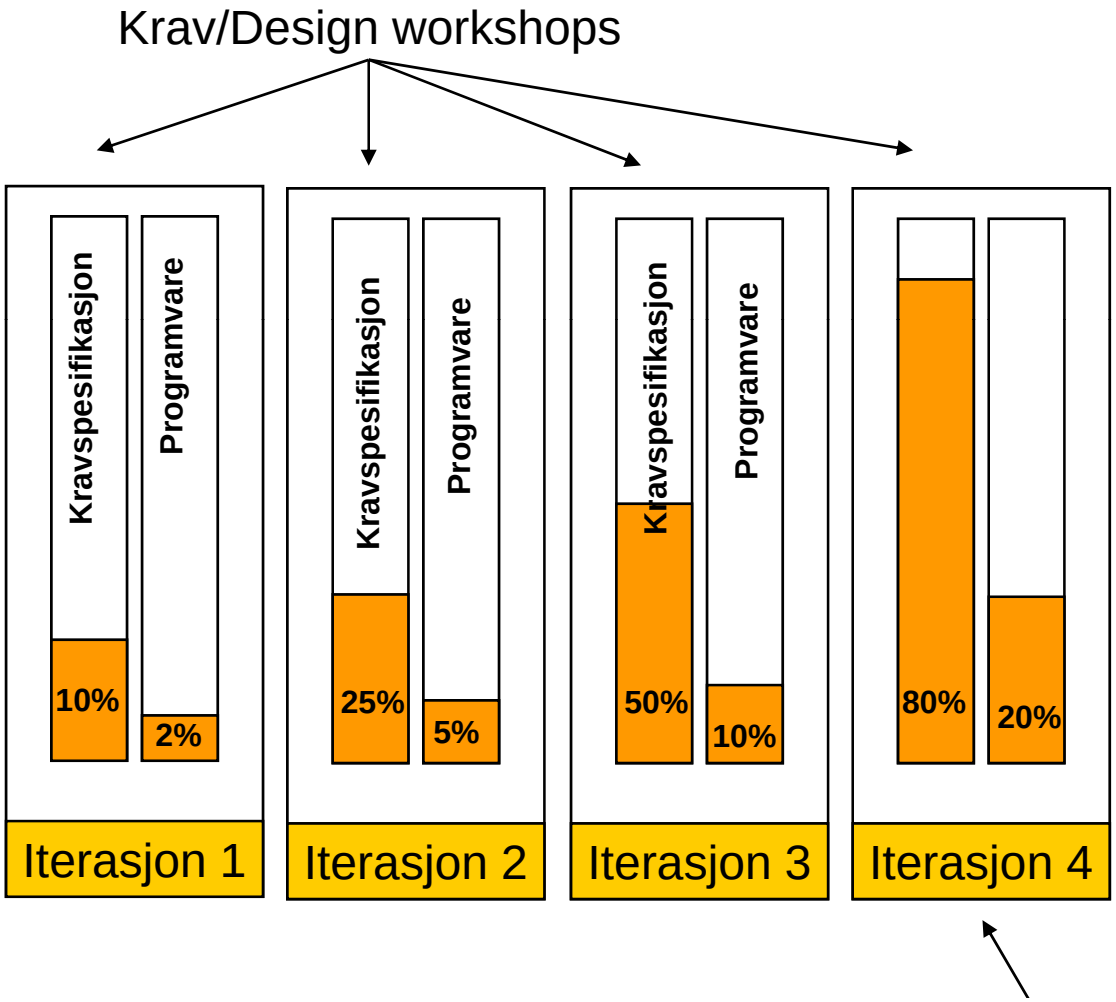
Eksempel


Iterasjonsplan
Analyse og Design
Programmering
Test
Workshop overordnet Kravinnsamling
Bruk resultat fra evaluering i forrige iterasjon
Kravene analyseres
Grov skisse til design
Estimering
Endelig prioritering av krav
Utviklere tester
Brukere tester
Beslutningspunkt
Er alt OK?

Eksempel på prioritert kravliste

	Item #	Description	Est	By
Very High				
	1	**Finish database versioning**	16	KH
	2	**Get rid of unneeded shared Java in database**	8	KH
	-	**Add licensing**	-	-
	3	Concurrent user licensing	16	TG
	4	Demo / Eval licensing	16	TG
		Analysis Manager		
	5	File formats we support are out of date	160	TG
	6	Round-trip Analyses	250	MC
High				
	-	**Enforce unique names**	-	-
	7	In main application	24	KH
	8	In import	24	AM
	-	**Admin Program**	-	-
	9	Delete users	4	JM
	-	**Analysis Manager**	-	-
	10	When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab	8	TG
	-	**Query**	-	-
	11	Support for wildcards when searching	16	T&A
	12	Sorting of number attributes to handle negative numbers	16	T&A
	13	Horizontal scrolling	12	T&A
	-	**Population Genetics**	-	-
	14	Frequency Manager	400	T&M
	15	Query Tool	400	T&M
	16	Additional Editors (which ones)	240	T&M
	17	Study Variable Manager	240	T&M
	18	Haplotypes	320	T&M
	19	**Add icons for v1.1 or 2.0**	-	-
	-	**Pedigree Manager**	-	-
	20	Validate Derived kindred	4	KH
Medium				
	-	**Explorer**	-	-
	21	Launch tab synchronization (only show queries/analyses for logged in users)	8	T&A
	22	Delete settings (?)	4	T&A

Krav kan utvikles over tid



- **1ste workshop: alle krav detaljeres opp til 10%**
- **Visjon for arkitektur**

Brukbare estimater får vi kanskjeikke før her

Fordeler med evolusjonære modeller

- Interessentene er tidlig med i planlegging og evaluering
- Evaluering av delmål underveis
  - Involverer interessentene som får prøve deler av systemet
- Støtter endringer underveis i utviklingen
  - Tilbakemelding fra interessentene mht. hva som fungerer/ikke fungerer benyttes til å planlegge videre utvikling
- Når evaluering og nødvendige endringer gjøres ofte nok har vi en rimelig sjanse for å oppdage feil tidlig
- Velegnet for hyppig risikoanalyse

Ulemper med evolusjonære modeller

- Krever selvstendige og proaktive prosjektmedlemmer
- Mindre grad av formalisme
  - Ser enkelt ut, men krever disiplin
  - Risiko for mindre vekt på kravanalyse og test
- Hovedvekt på funksjonalitet (bygge rett system)
  - Lett å glemme tekniske krav
  - Mangler gode løsninger for arbeid med arkitektur
  - Lite oppmerksomhet på vedlikeholdbarhet

Evolusjonære modeller

Ville Oslo Sporveier spart penger ved å installere og teste billettautomatene og tilhørende programvare på noen få stasjoner istedenfor på alle?

Instanser av evolusjonære modeller

- Spiralmodellen
- Rational Unified Process (RUP)
- Extreme Programming (XP)
- Scrum

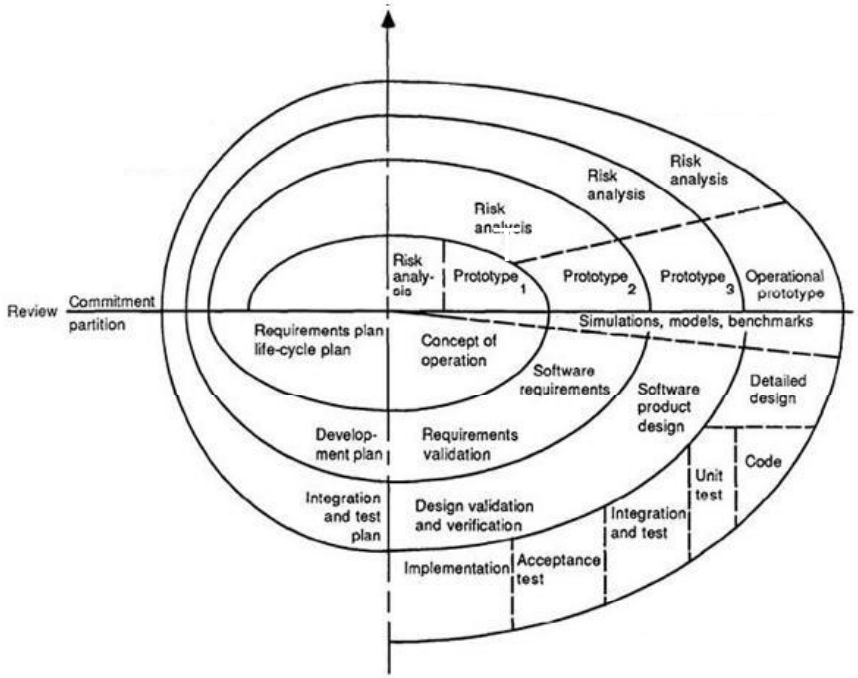
Spiralmodellen betraktes mer som en referansemodell (meta-modell) enn en detaljert og godt beskrevet utviklingsmodell.

Merk at objektorientering og strukturert utvikling ikke betraktes som egne utviklingsmodeller i disse forelesningene.

Spiralmodellen

Fire kvadranter:

1. Planlegging
2. Risikoanalyse
3. Design, programmering, etc
4. Evaluering av kunde

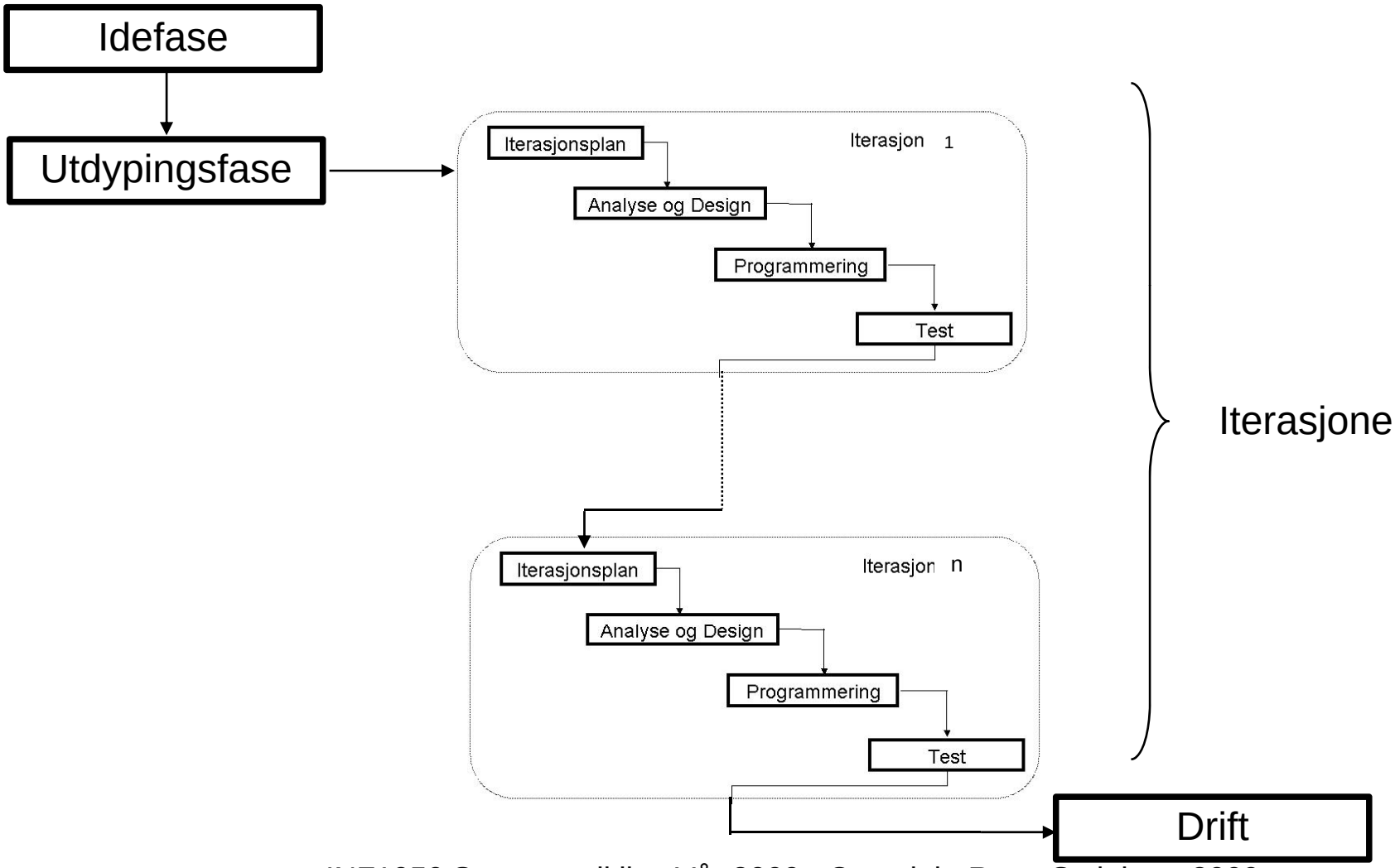


Spiralmodellen var en av de første modellene som eksplisitt innførte risikoanalyse:
Hva kan gå galt, med hvilken sannsynlighet, og konsekvens?

Rational Unified Process: Faser

- Idefase:
  - Overordnet målsetting, behovsanalyse, budsjett, prosjektplan
  - Start innsamling av funksjonelle krav og modellering av use cases
- Utdypningsfase:
  - Fortsett med use cases og tekniske krav
  - Start design av arkitektur, lag arkitektur prototype
  - Ferdigstill prosjektplanen
- Iterative konstruksjonsfaser: (design-programmering-test)
- Overgang til drift: stabilisering, ferdigstilling

RUP


Idefase
Utdypingsfase
Iterasjonsplan
Iterasjon 1
Analyse og Design
Programmering
Test
Iterasjonsplan
Iterasjon n
Analyse og Design
Programmering
Test
Iterasjoner 1...n
Drift

Rational Unified Process

- RUP er en arkitektursentrert prosess
- Bygger på objektorienterte utviklingsprinsipper
- UML modellering er sentralt, basert på 4+1 modell:
  - Logisk (funksjonelt) perspektiv (klasse diagrammer)
  - Prosess nivå (intern koordinering i systemet - parallellitet, synkronisering)
  - Fysisk nivå (hvordan programvare samvirker med maskiner og nettverk)
  - Utviklingsnivå (programmereren perspektive)

RUP er en sterkt *modell*-drevet prosess der mye fokus legges på å bygge arkitektur (UML) modeller. Ulempen med dette er som ved fossefallsmodellen. Det er vanskelig å benytte diagrammer til å vurdere om systemet dekker brukernes behov. Overdreven utvikling av diagrammer er kostbart og lite effektivt.

Lettvektsmetoder

- Lettvektsmetoder også kalt smidige (agile) metoder er en egen gruppe av utviklingsprosesser
- Vektlegger mindre formalisme (krav til leveranser) og mindre dokumentasjon.
- Oppfordrer i stedet til direkte muntlig kommunikasjon mellom prosjektdeltagere
- Noen inneholder programmeringsnære teknikker, andre ikke
- Det finnes mange forskjellige navngitte metoder, Scrum, og Extreme Programming (XP) er mye brukt
- Uenighet om RUP tilhører denne klassen

Lettvektsmetoder: Mantra

- Individer og kommunikasjon fremfor prosesser og verktøy
- Fungerende programvare fremfor omfattende dokumentasjon
- Samarbeid med kunde fremfor kontraktsforhandlinger
- Endringsvillighet fremfor å følge prosjektplanen

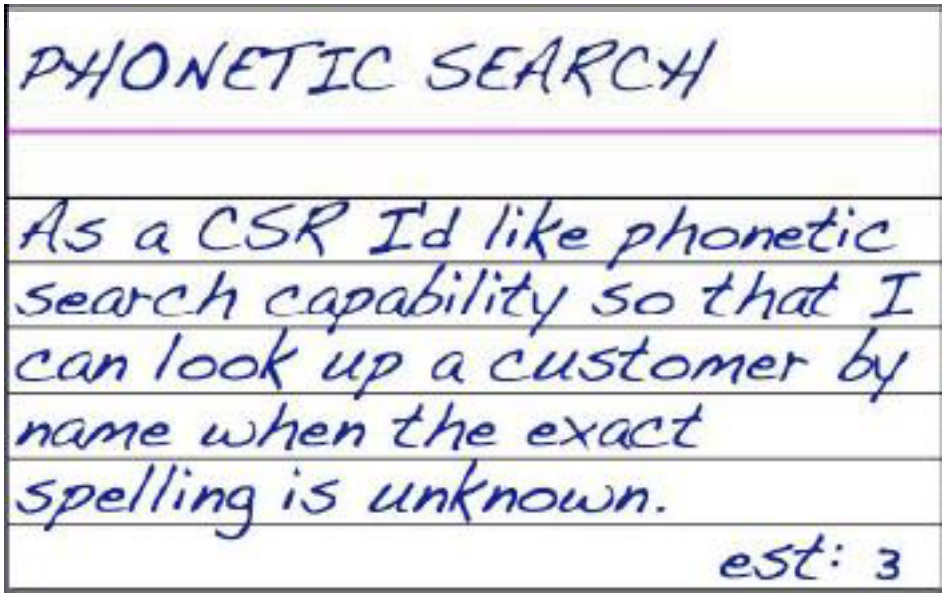
Extreme Programming (XP)

- Fokus på programmering, test og tilhørende teknikker
  - Parprogrammering
  - Refaktorering (ombygging etter behov)
  - Testdrevet utvikling (noen tester lages før og under programmering)
- Få krav til spesifikasjon og planlegging
  - Benytter små *bruksscenarier (user story)*
- Svært korte iterasjoner (1-3 uker)
- Interessentene (bruker) integrert i prosjektorganisasjonen
- Interessentene prioriterer utviklingsoppgaver for hver iterasjon
- Interessentene vurderer resultatene
- Lite formalisme i metoden, krever derfor mye disiplin

Extreme Programming faser

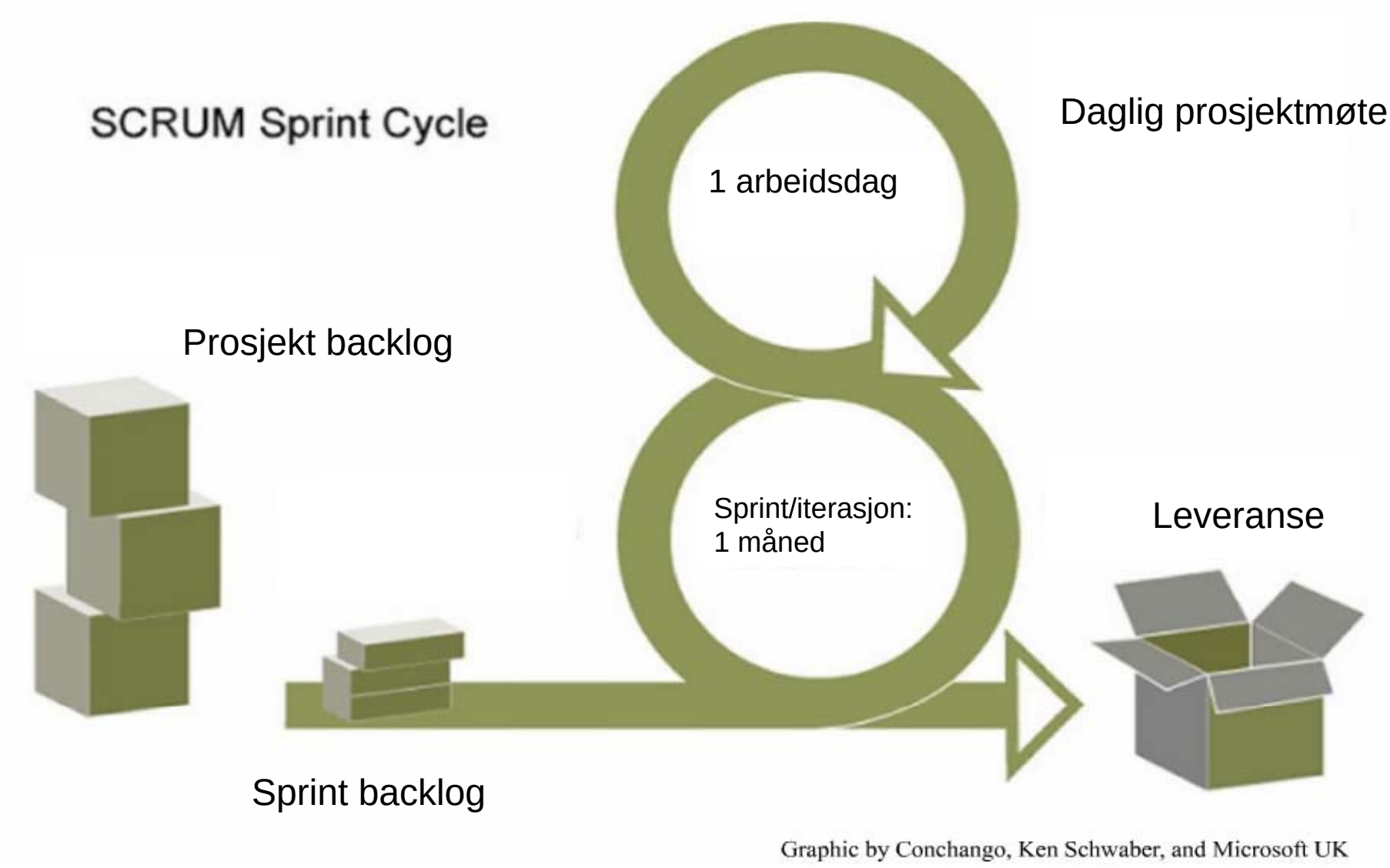
- Foranalyse: lønnsomhetsvurdering ved behov
  - overordnede bruksscenarier
- Iterasjonsplan: identifiser de viktigste bruksscenariene
  - Skrives av bruker/kunde på små enkle kort
  - Kortene estimeres og splittes hvis de er for store
- Analyse og Design: svært enkel tilnærming
  - Design kan ofte gjøres på tavle eller små kort (CRC)
  - Noen designoppgaver vil kreve utvikling av en prototype
- Programmering
  - Programkoden inkluderer kode som benyttes i daglig testing
  - Tidligere utvilet programkode og arkitektur bygges om ved behov
- Test
  - Bruker/kunde tester resultatet

Extreme Programming Krav
Eksempel på bruksscenarie



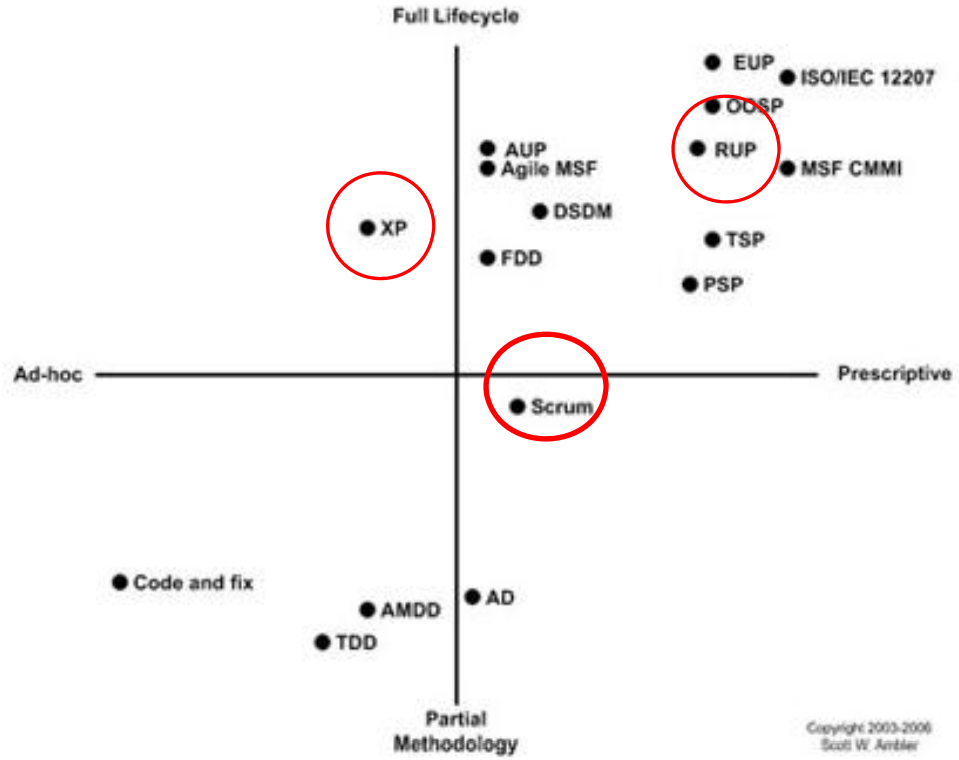
Følger mønsteret: Rolle vil utføre noe for å oppnå et gitt resultat

Scrum


SCRUM Sprint Cycle
1 arbeidsdag
Daglig prosjektmøte
Prosjekt backlog
Sprint/iterasjon:
1 måned
Leveranse
Sprint backlog
Graphic by Conchango, Ken Schwaber, and Microsoft UK

Lettvekstmetoder

Det finnes mange lettvektsmodeller med varierende grad av formalisme:



Prosjektarbeid

Læringsmål

- Forstå prosjektarbeidets rolle i systemutvikling
- Kjennskap til de viktigste delene av prosjektorganisasjonen
- Forstå hovedoppgavene for prosjektlederen
- Kunnskap om hvordan planlegge, overvåke og styre et prosjekt
- Kunnskap om suksessfaktorer

Hva er et prosjekt?

- En engangsoppgave som ikke er utført tidligere
- Skal lede til et bestemt resultat
- Krever ulike typer (tverrfaglige) ressurser
- Begrenset i tid
- Organiseres ofte utenfor virksomhetens linjeorganisasjon

Prosjektarbeidets rolle i systemutvikling

- Formålet med prosjektarbeid er å organisere, kontrollere og styre prosessen.
- De ulike utviklingsmodellene foreslår forskjellige tilnærminger til hvordan dette skal gjøres

Hvorfor er prosjektarbeid viktig?

- Planlegging:
  - Hvordan kan vi effektivt utnytte ressursene i prosjektet?
- Oppfølging:
  - Utføres arbeidet etter planen?
  - Er det avvik mellom plan og fremdrift?
- Korreksjon:
  - Hvilke endringer er nødvendig?
  - Hvilke tiltak må gjennomføres for å levere med rett kvalitet uten å overskride budsjett eller leveransedato?

Hovedmål: Til enhver tid avklare om prosjektmålene er oppnålig innenfor prosjektets rammer (kostnader og krav)

Prosjektorganisasjonen

- Styringsgruppen
- Prosjektlederen
- Prosjektstab
- Gruppeledere
- Prosjektmedlemmer
- Ekstern kvalitetssikring

Mindre prosjekter har normalt ikke prosjektstab, gruppeledere, eller ekstern kvalitetssikring

Styringsgruppen

- Representerer normalt de økonomiske interessentene
- Ansvar for å forvalte prosjektets målsetting
- Styrer prosjektet på overordnet nivå
- Godkjenner prosjektplaner og endringer
- Håndterer problemsaker som prosjektleder ikke kan løse
- Koordinerer linjeorganisasjon og prosjekt

Prosjektlederen

- Daglig ledelse av prosjektet
- Ansvar for at prosjektet følger fastsatt plan:
  - Utarbeide og endre prosjektplan
  - Organisere ressurser, tildele oppgaver
  - Kontrollere resultat og fremdrift
  - Identifisere avvik
  - Foreta justering av planen iht. fremdriften
- Rapporterer til styringsgruppen

Viktige faktorer i planleggingen

- Kostnadsramme
- Tidsramme
- Personalramme (antall prosjektdeltagere og kompetanse)
- Utstyrsramme (maskiner, programvare, nettverk, etc)
- Krav til leveransene
- Offentlige krav (lover, retningslinjer, etc)
- Produksjonstekniske krav

Hovedelementer i prosjektplanlegging

- Identifisere og planlegge prosjektets mål og delmål
- Prioritere oppgaver
- Estimere arbeidsomfang
- Identifisere aktiviteter (arbeidsoppgaver)
- Beslutte start- og slutt dato for aktivitetene
- Holde oversikt over avhengigheter mellom aktivitetene
- Beslutte hvem skal utføre aktivitetene
- Planlegge hvordan resultatet skal evalueres
- Planlegge etablering og drift av produksjonsutstyr

Merk at overordnet planlegging gjerne gjøres sammen med interessentene som gjerne beslutter de overordnede krav til planleggingen

Milepæler

Tydelige og målbare målsettinger er nødvendig for at prosjektet skal lykkes:

- Milepæler beskriver hvilke (del)mål som skal oppnås
- Milepælene fungerer som kontrollstasjoner underveis
- Faser i utviklingsprosessen blir normalt overordnede milepæler

Eksempler på milepæler i en iterasjon

- Når foranalysen er utført
- Når iterasjonsplanen er godkjent
- Når alle planlagte mål er spesifisert og designet
- Når programmeringen av alle funksjoner er ferdig
- Når test og evaluering er utført

Aktivitetsplan

Aktivitetsplanen er den mest detaljerte delen av planen. En aktivitet er en avgrenset del av arbeidet og inneholder følgende:

- Navn
- Arbeidsomfang (estimat) for hver aktivitet
- Tidspunkt for start og slutt
- Hvilke ressurser som skal utføre aktiviteten (inkl. last)
- Avhengigheter og rekkefølge i forhold til andre aktiviter som skal utføres

Estimering

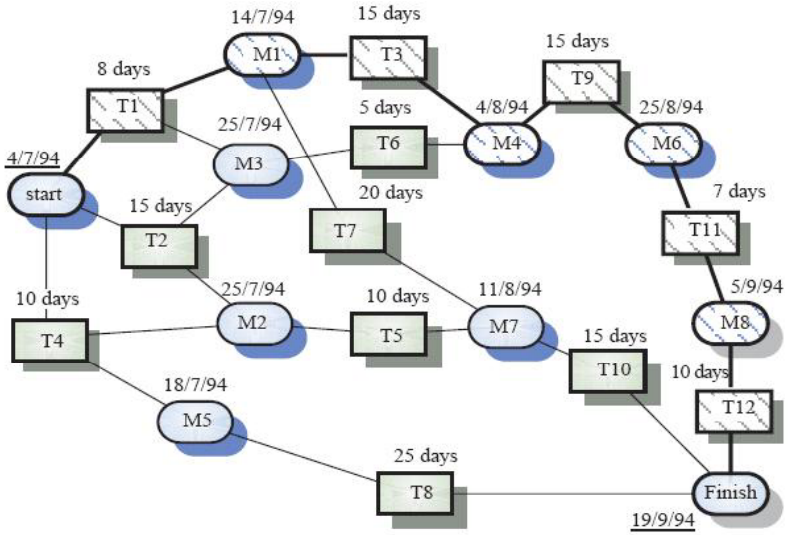
For å lykkes må det være samsvar mellom hvor mange arbeidstimer som kreves og arbeidskapasiteten. Estimering innebærer å forutsi hvor mye arbeid som kreves for å fullføre en oppgave.

- Estimering er i beste fall kvalifisert gjetning
- Krav og forutsetninger endres underveis
- Historisk sett har vi lav sannsynlighet for å treffe

Nettverksdiagram

Nettverksdiagram (prosjektnettverk) beskriver mulige gjennomføringer av aktivitetene. Nettverksdiagrammet bestemmes ved:

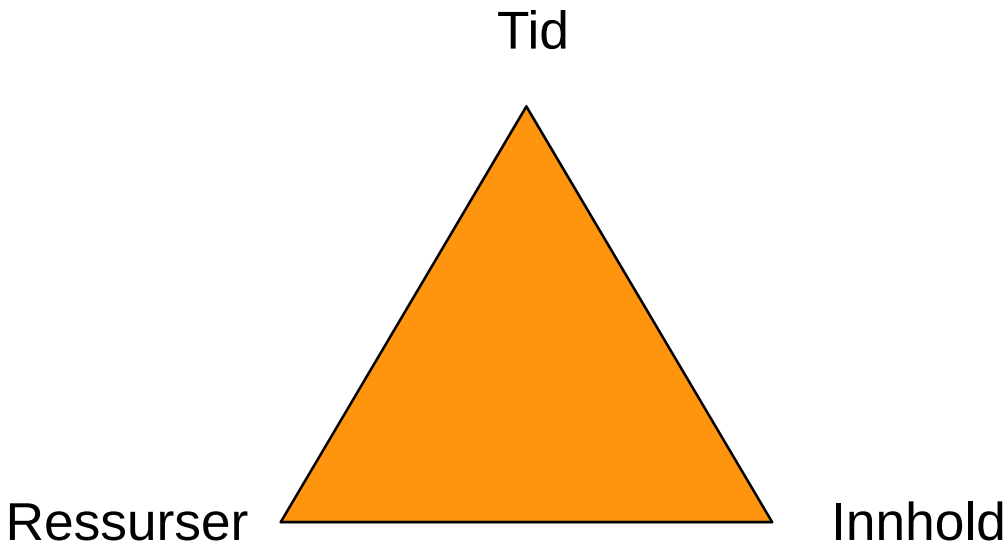
- Hver aktivitet defineres som en enhet i diagrammet
- Mulige rekkefølger angis med piler mellom to aktiviteter
- Start tidspunkt og tidspunkt for ferdigstilling bestemmer diagrammets mulige veier



Ressursplanlegging

- Når vi har definert nettverksdiagrammet kan vi begynne å tilordne ressurser
- Ingen jobber 100%, sykdom, ikke planlagte gjøremål må iberegnes
- Kompetanse og interesse må vurderes
- Tilgjengelig kapasitet vil bestemme rammen

Kritiske faktorer



Vi har sett at sannsynligheten for å kunne levere som opprinnelig planlagt er svært liten. Muligheten til å endre minst en av disse faktorene er normalt nødvendig for å lykkes.

Når leveransedato er prioritert

Når leveransedatoen ikke kan flyttes benyttes en teknikk som kalles *timeboxing.* Normalt må innhold justeres etter budsjett og ressurser.

- Det er fase-bestemte milepæler som gjerne definerer timeboxen
- Start og slutt dato kan ikke endres
- Interessentene gir alle krav en unik prioritet
- Utviklingen starter med viktigste krav og fortsetter med mindre viktige til tiden er oppbrukt

Selv om tidsplanen holder kan ikke timeboxing garantere at systemet vil inneholde et minimum av nødvendig funksjonalitet!

Formål med oppfølging og korreksjon

- Oppfølging
  - Hvordan er framdriften i henhold til planen?
  - Hvilke aktiviteter holder planen, hvilke er forsinket?
  - Har noen problemer?
  - Har vi oppdaget noe uforutsett?
  - Er det behov for endringer?
- Korreksjon
  - Har vi nok ressurser?
  - Er alle oppgavene riktig fordelt?
  - Trenger noen hjelp?
- Bør vi revidere planen?

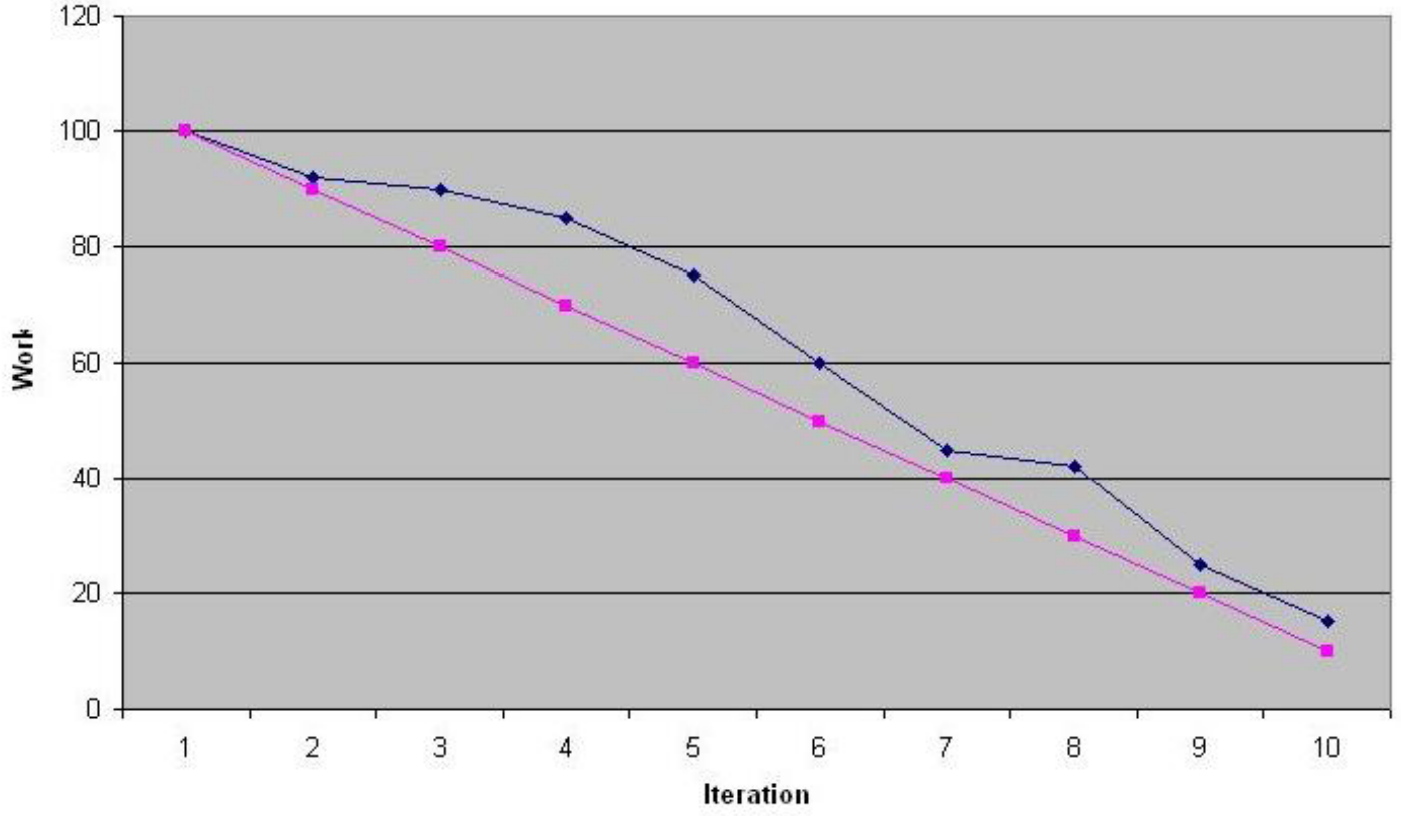
Oppfølging av fremdrift

Budsjett	100t	Opprinnlig budsjett
Plan	120t	Antatt sluttresultat
Utført	50t	Utført til nå
Gjenstår	70t	Videre arbeid
Resultat	120%	Antatt overskridelse

Dersom det er en uke igjen før denne oppgaven skal avsluttes og oppgaven bare kan utføres av en person vet vi at vi riskerer å bli forsinket.

Oppfølging av fremdrift

Eksempel på en visuell fremstilling av fremdrift basert på planlagt gjenstående (fiolett) og faktisk gjenstående (blå)



Oppfølging av fremdrift

Denne tilnærmingen krever at:

- Alle rapporterer hvor mange arbeidstimer som har medgått pr aktivitet
- Alle aktiviteter re-estimeres jevnlig (f. eks. 1 gang i uken)
- Prosjektleder oppdaterer planer og lager en antagelse om hvordan prosjektet vil utvikle seg

Alternativ metode

- Alle oppgaver i prosjektet deles inn i så små oppgaver at de hver tar en fast tidsenhet, f. eks. 1 dag
- Hver person får en oppgave pr dag av prosjektleder
- Dersom gårsdagens oppgave ikke er ferdig dagen etter må korrektive tiltak iverksettes

Timebasert fremdriftskontroll forutsetter nærmest at prosjektet bygger den riktige løsningen på riktig måte. Dette skjer implisitt gjennom antagelsen om at planlagte aktiviteter er de riktige aktiviteten som må utføres for å lykkes.

Risikostyring

Risiko = Sannsynlighet x Konsekvens

Målsetting med risikostyring er å tenke fremover. Forsøk å identifisere hva som kan gå galt, vurdere sannsynlighet og konsekvens. Er det noe vi kan gjøre nå for å dempe risikoen?

- Løsning
  - Har vi kontakt med de riktige interessentene?
  - Har vi forstått kravene og er de tilstrekkelig komplette?
- Ressurser
  - Har vi nok ressurser, har de rett kompetanse, har de rett utstyr?
- Kommunikasjon
  - Klarer vi å unngå misforståelser? Hvordan unngå å glemme krav?

Suksessfaktorer

Standish CHAOS study 1994:

Årsaker til problemer:

- Ufullstendige krav
- Manglende brukerinvolvering
- Manglende ressurer

Årsaker til sukess:

- Utbredt brukerinvolvering
- Støtte fra ledelsen
- Tydelige krav

Det er derfor nyttig å følge en utviklingsprosess som understøtter disse anbefalingene

Suksessfaktorer

Det er helt essensielt å bygge rett system. Det krever:

- Tett samarbeid og kommunikasjon med sluttbrukere
- Sluttbrukerne/kunde må få se og prøve deler av systemet underveis i prosjektet
- Korte iterasjoner (kort nok til at vi har råd til å feile)
- Små og oversiktlige prosjekter og prosjektgrupper

Teknologi er sjelden nevnt som en viktig suksessfaktor...