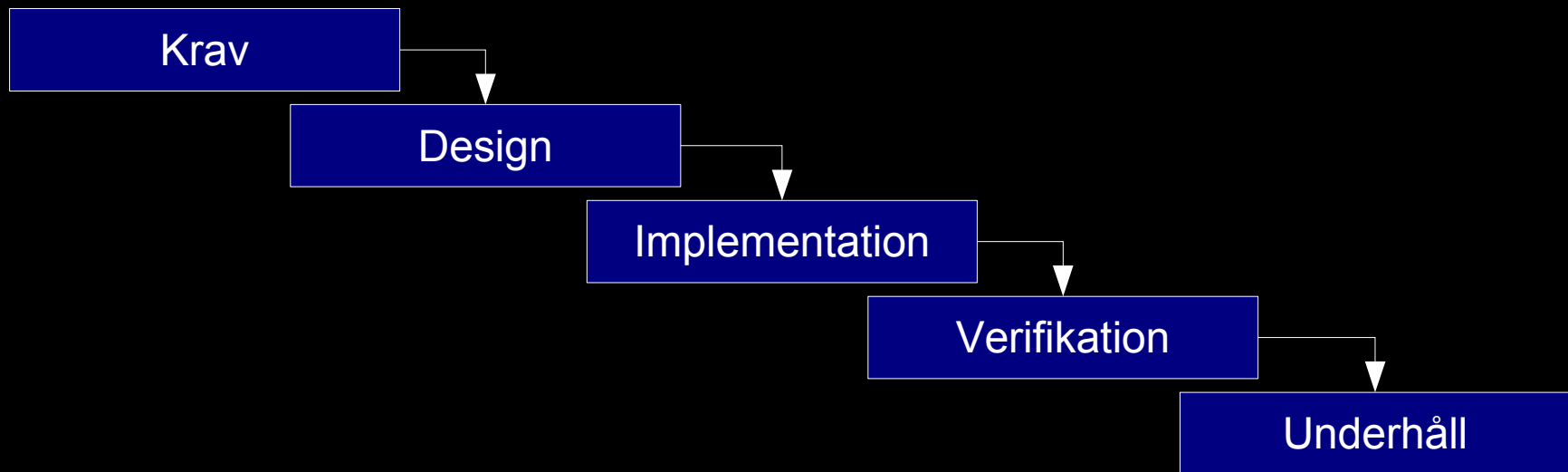


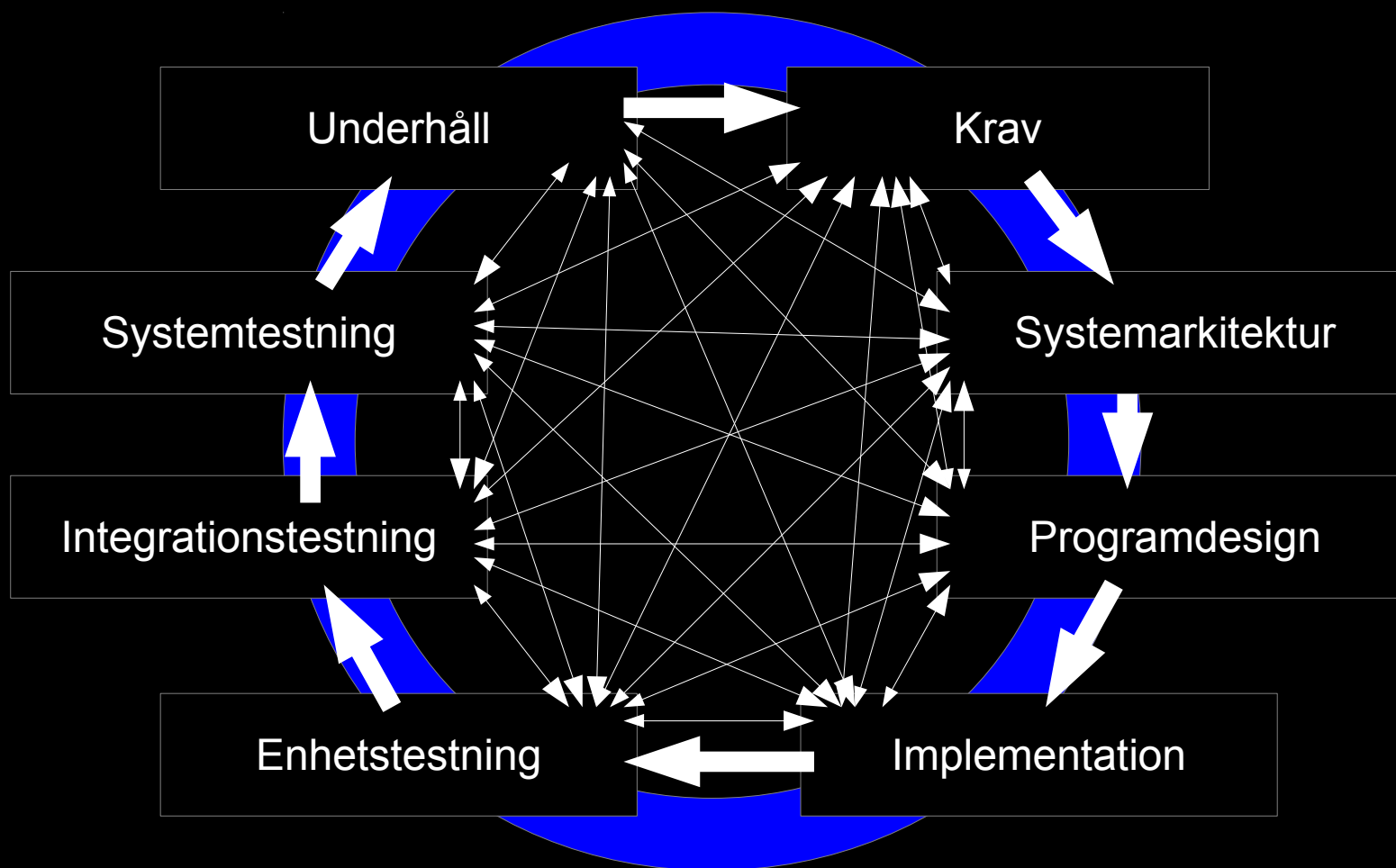
TNM094 – Medietekniskt kandidatprojekt

Metodik för systemutveckling
Utvecklingsmodeller

Systemutveckling



Systemutveckling



En explicit utvecklingsprocess

- Sekvens av aktiviteter
 - en samling organiserade aktiviteter
 - sammansatta för att uppfylla målen med låg OH
- Guidande principer och ramar
 - vad ska göras, när ska det göras och hur
 - vanliga faser, milstenar och leverabler

Viktiga aspekter för metodiken

- Skapa en gemensam förståelse för arbetet
- Skapa en överblick
 - för att hitta inkonsekvens
 - för att hitta redundans
 - för att hitta saknade moment
- Process-anpassning
 - justera processerna för det specifika projektet
 - team, schema och mål
 - teknik, funktioner och omfattning
 - anpassa och specialisera
 - men vissa moment bör inte ändras!

Viktiga principer

- Vi vill ha
 - **Kontroll** över projektet
 - **Flexibilitet** vid förändrad kunskap eller förutsättningar
- Andra principer
 - Big / Rough Design Up Front
 - Analys-paralys

Kontroll

- Kontroll över arbetet och resultatet
 - Vad ska göras, när och hur?
 - Vilka funktioner är klara när och till vilken kostnad?
 - För kunden, utvecklarna och ledningen
 - Mycket kunskap tidigt i projektet
- Kontroll kostar
 - I resurser
 - I flexibilitet

Flexibilitet

- Vi vill kunna anpassa oss under utvecklingen
 - Förändringar i förutsättningarna
 - Felaktig planering p g a okunskap
 - Arbete ger kunskap och ny information uppstår löpande
 - Opålitlig eller okunnig kund
- Svara på förändringar
 - Processen kan tillåta förändring under utvecklingen
 - Kan också justera processen för att passa situationen

* Design Up Front

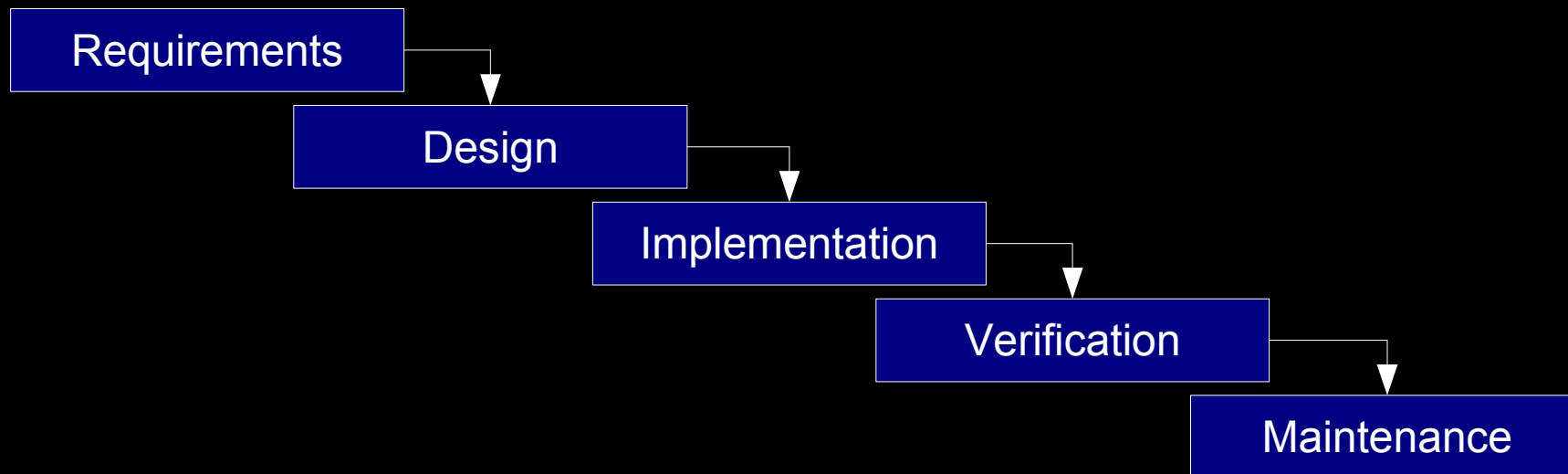
- Kostnad för planering är lägre än för att fixa senare
- Planering av vad som inte görs är onödig kostnad
- Hur noga ska man planera?
 - BDUF – generell plan över hela systemet
 - RDUF – grov plan över vad man fokuserar på
 - Emergent design – äsch, det löser sig
(bästa design kommer av sig självt)

Analys-paralys

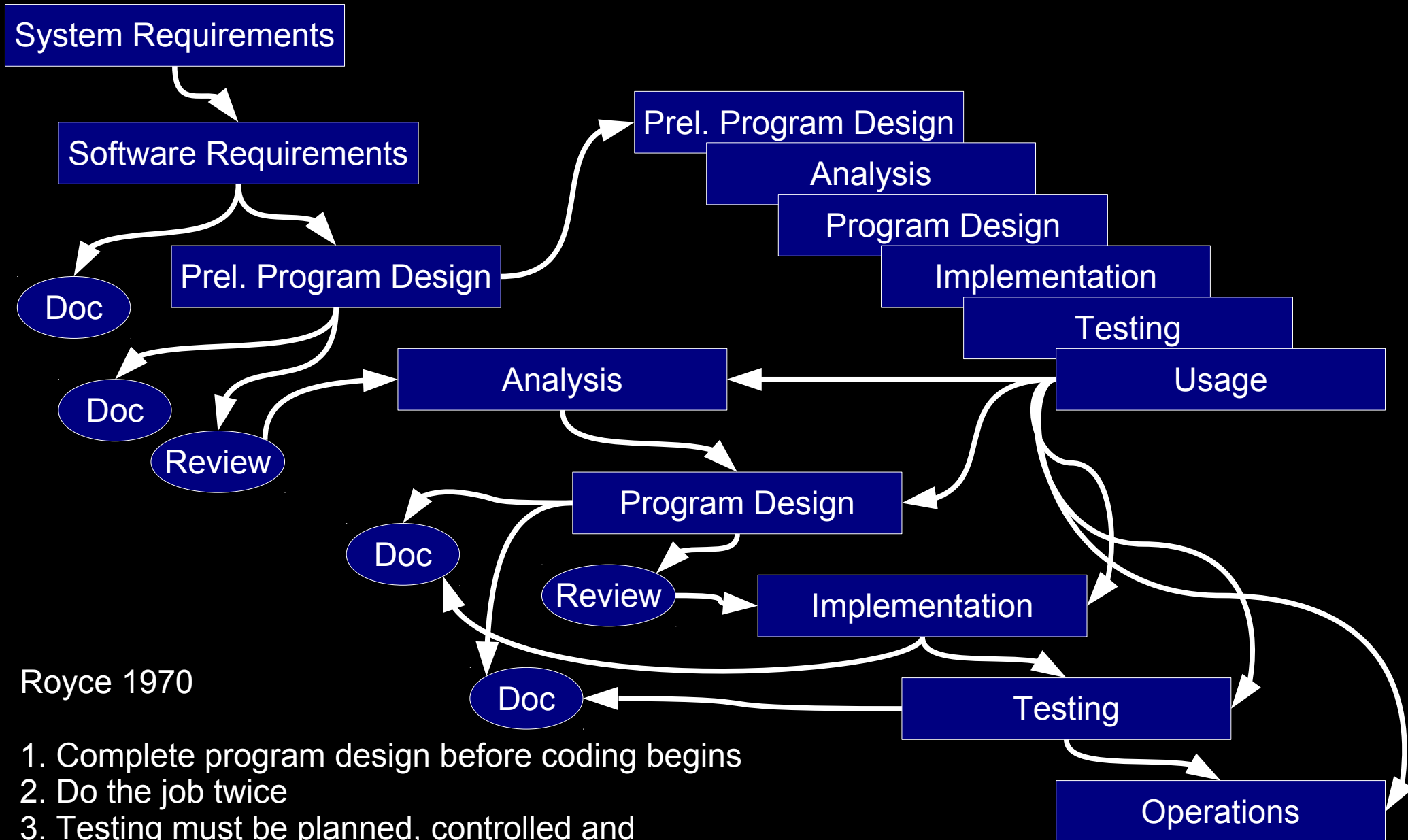
- Planering leder lätt till mer planering
 - Frågor leder lätt till fler frågor
 - Göra och testa leder till mer kunskap och svar
- Hur hittar vi rätt balans?
 - När ska man sluta planera?
 - När är planen “tillräckligt bra”?
- Enkla lösningen
 - Begränsa analysen i tid!

Vattenfallsmodell

- Steg-för-steg fungerar inte
 - Kraven visar sig ofta underspecificerade
 - Designen visar sig inte fungera vid implementation
 - Implementationen visar sig felaktiv vid testning
 - Mycket är okänt vid projektets start
- ... men används inte heller



Modifierad vattenfallsmodell



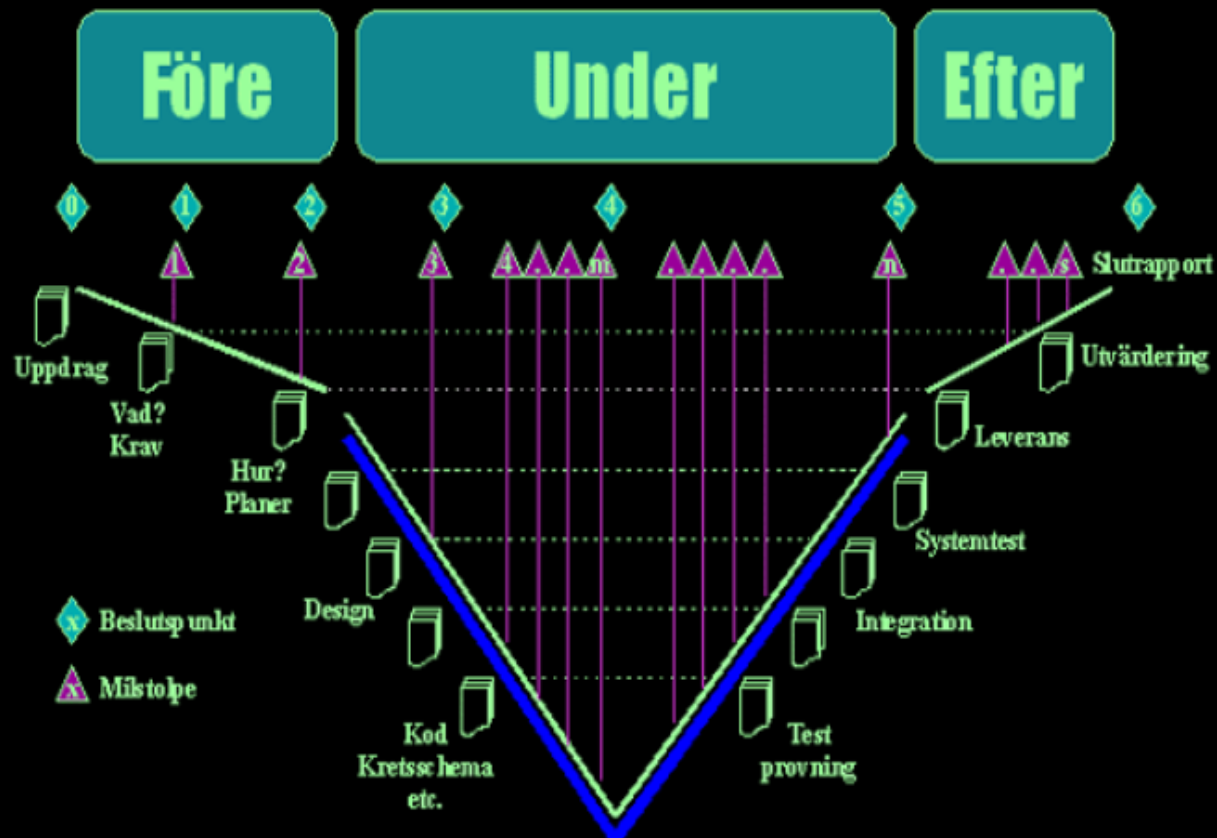
Royce 1970

1. Complete program design before coding begins
2. Do the job twice
3. Testing must be planned, controlled and monitored
4. Involve the customer

[Royce 1970]

Linjära modeller

- fungerar bra vid låg nivå av osäkerhet
- V-model
- LIPs – Lätt Interaktiv Projektstyrning



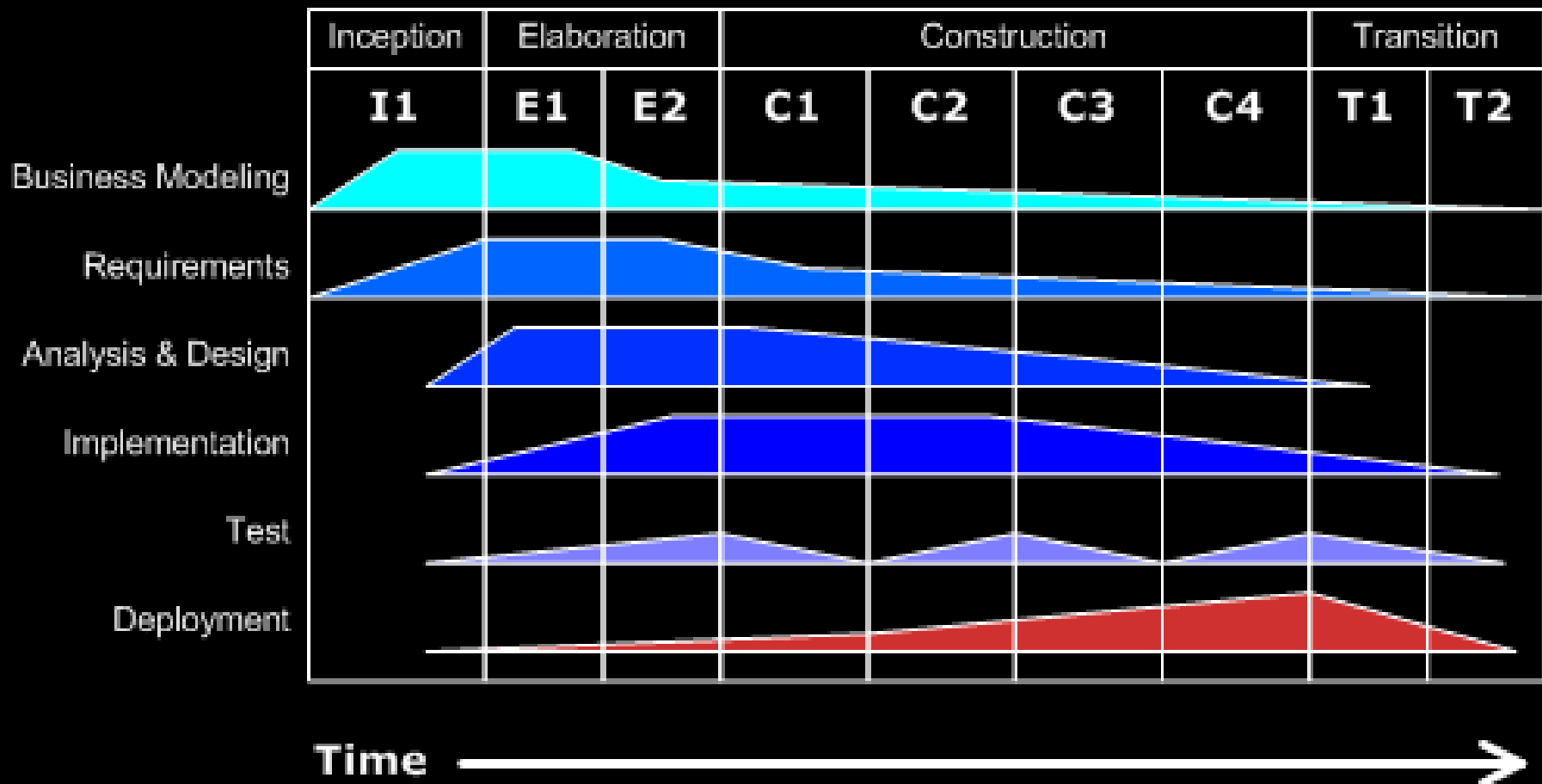
The Unified Process (UP)

- En uppsättning principer och processer
- Anpassas till organisation och projekt
- Faser
 - Inception, Elaboration, Construction, Transition
(Påbörjande, fördjupning, konstruktion, överlämning)
- Nyckel-arbetsflöden
 - Kravhantering, analys, design, implementation, testning
- Hårt planerad iterativ utveckling
- Utvidgningar och vidareutvecklingar
 - OpenUP (OUP), Rational UP (RUP), Enterprise UP (EUP), etc.
 - AgileUP (AUP), Disciplined Agile Delivery (DAD)

The Unified Process (UP)

Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.



Agil utveckling

- Värderar individer och interaktion
över processer och verktyg
- Värderar fungerande programvara
över omfattande dokumentation
- Värderar kundsamarbeten
över kontraktsförhandlingar
- Värderar att svara på förändring
över att följa en plan

Agil utveckling

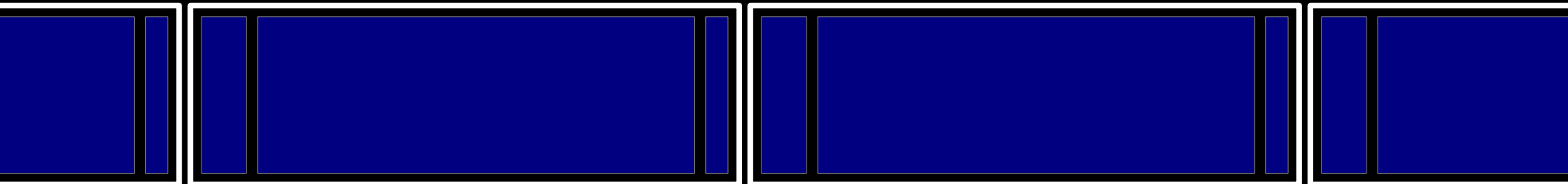
- Viktigt att tänka på
 - mer robust vid förändringar i projektet
(inte nödvändigtvis vid förändring i processen)
 - passar bättre för små utvecklingsteam
(och mindre projekt)
 - “värdera över” *inte* “istället för”

Agil utveckling

- Viktigt att tänka på
 - Refaktorering ersätter detaljerad planering
 - refaktorering tar tid utan att leda till funktionalitet
 - vissa projekt utelämnar tid för refaktorering
 - refactoring krävs för att undvika dålig kodbas
 - Dokumentation är viktigt för framtida arbete
 - dokumentering tar tid utan att leda till funktionalitet
 - vissa projekt utelämnar tid och rutiner för dokumentering
 - brist på dokumentation gör framtida utveckling och underhåll dyrt
 - Agil utveckling hos leverantör
 - passar bäst för öppna kontrakt, tidsdebitering
 - svårt att använda vid upphandlingar

Scrum

- Krav hanteras i en produkt-backlog
 - med *poster* som beskriver efterfrågade beteenden
- Koordineras i *Scrum*
 - Dagliga, korta möten mellan utvecklarna i teamet
- Utveckling i *Sprintar*
 - Utvecklar utvalda poster till en fungerande produkt



Scrum-teamet

- Scrum-mästare
 - Säkerställer att Scrum förstås och efterlevs
- Produktägare
 - Kundens röst
 - Ansvarar för att värdemaximera arbetet
 - Ansvarar för produktbackloggen
- Utvecklingsteamet
 - en liten grupp med självorganiserande utvecklare
 - styr själva *hur* de ska utföra arbetet

Sprint

- Utvecklingen sker i sprintar
 - en kort tidsperiod, mellan en vecka och en månad
 - tillägnad ett urval av backloggen
- Inleds med sprint-planering
 - produktägaren förhandlar med teamet om *vad* som ska göras
 - teamet planerar *hur* det ska genomföras (omvandlar *poster* till *uppgifter*)
- Avslutas med sprint-genomgång
 - teamet presenterar resultatet för produktägaren
 - alla poster markeras som *färdig* eller inte
- Följs av en sprint-återblick
 - teamet diskuterar organisation och verktyg
 - förbättrar team, arbetssätt och infrastruktur



Scrum-scrum

- Internt samverkansmöte
 - kort och koncist, dagligt möte
 - alla utvecklare diskuterar aktuell status
 - ofta stående möten – för korta diskussioner
 - kan göras hierarkiska (scrum of scrums)



(c)2005 Iulianu @ Wikipedia, CC-AS

[Schwaber][Pfleeger]

Metodiker kombineras

- Som en låda i en låda
 - Hög- och lågnivå-metodiker kombineras
 - Test Driven Development (TDD)
 - Agile Modelling (AM)
 - Adaptive Software Development (ASD)
 - Välj, kombinera och justera
 - Varje projekt är unikt