

Mjukvaruprojekt Inception-fasen

1DV404, HT14

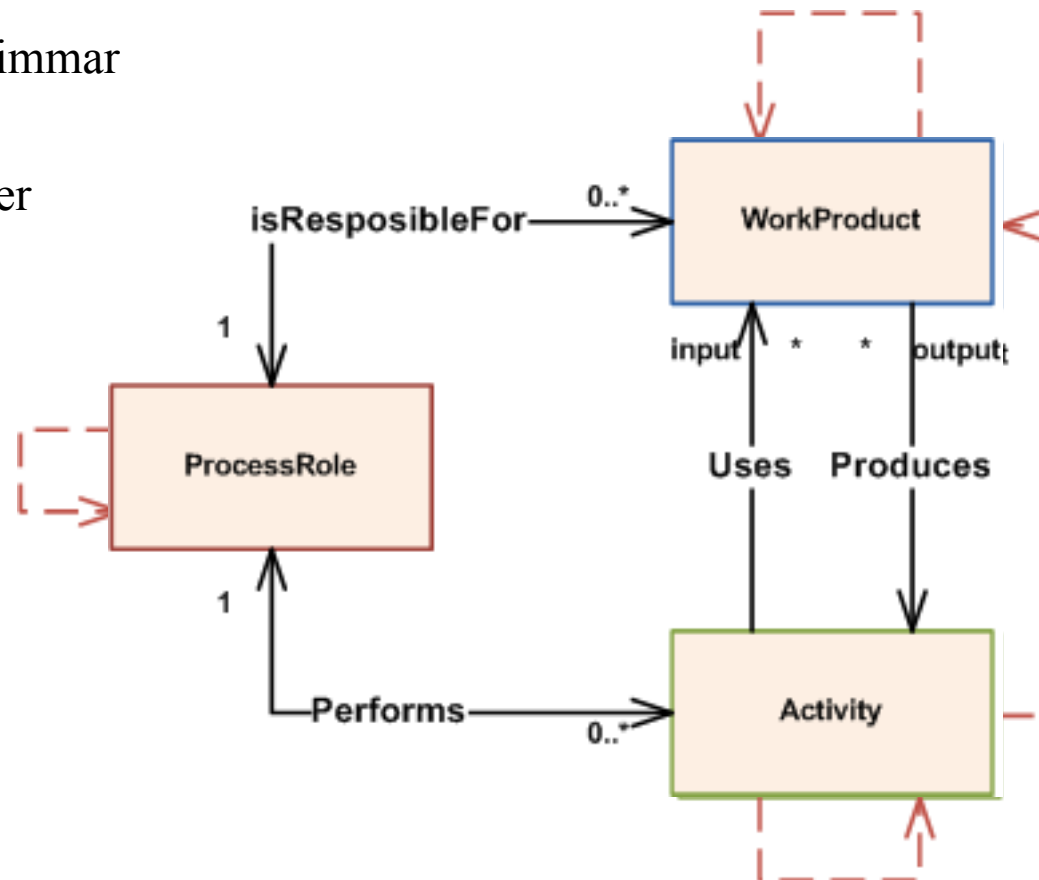
Jesper Andersson

Kap 5, 6, 7



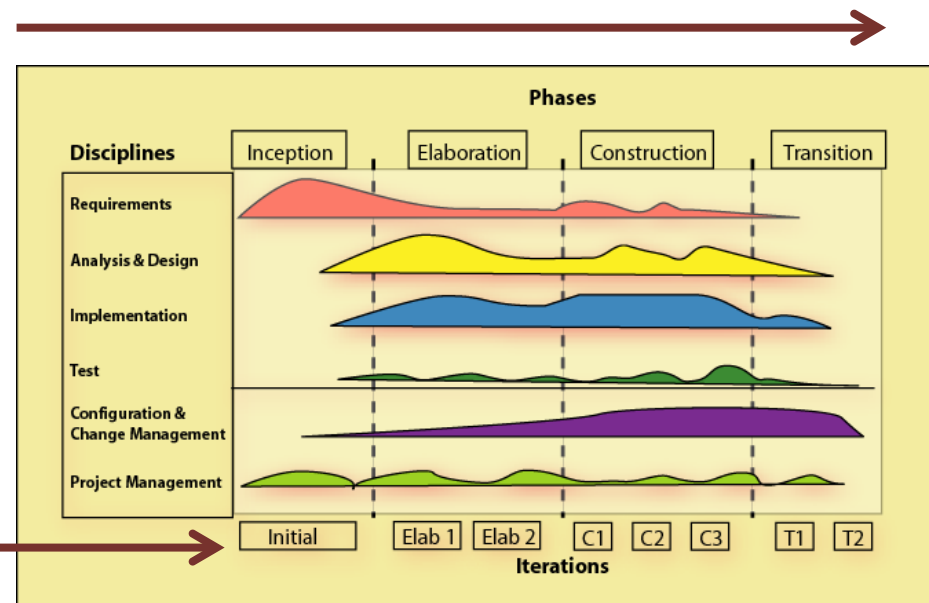
Hur organiserar vi arbetet i en processmodell.

- ✓ Roller – Resurser – Mänskliga
- ✓ Aktiviteter
- ✓ WorkProducts – Artefakter

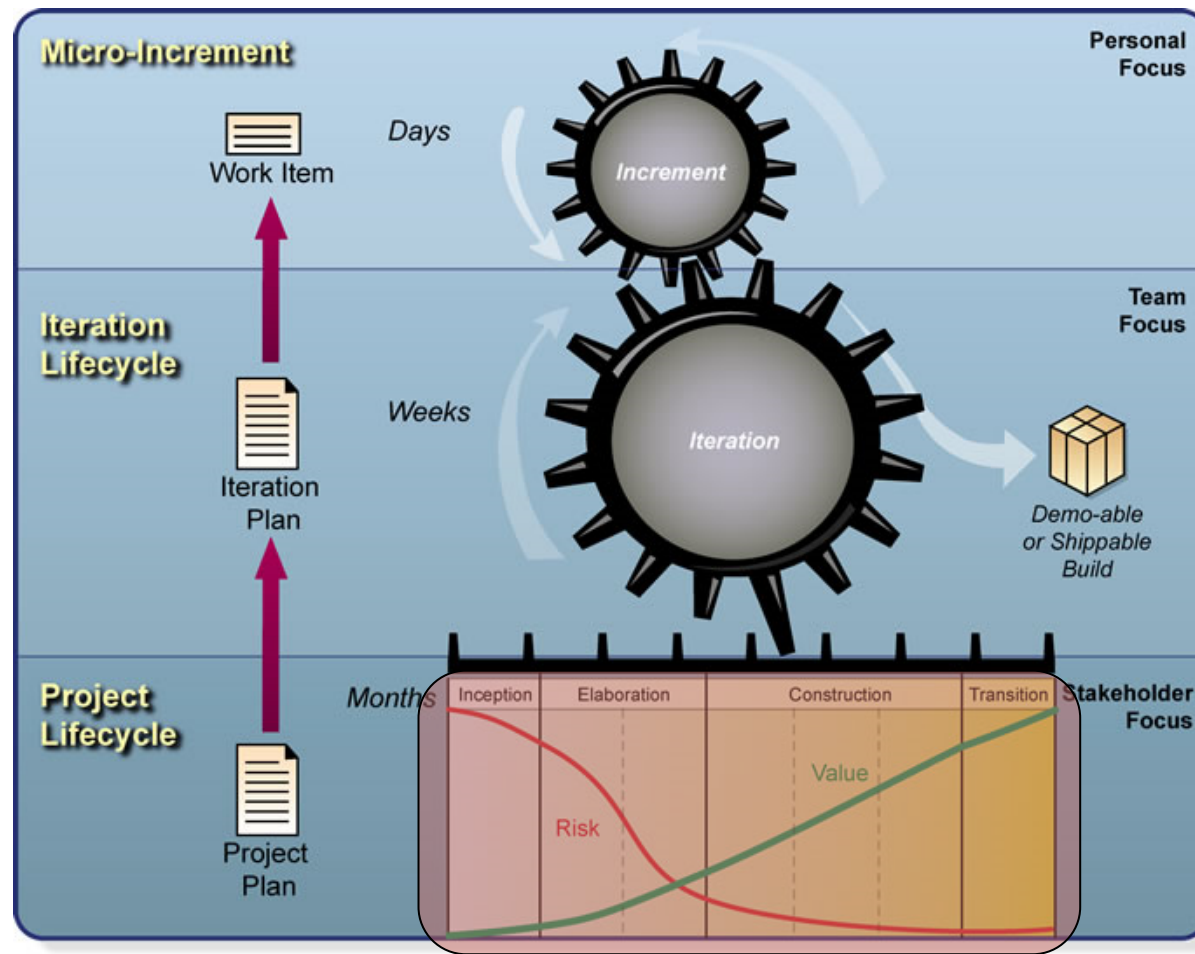


UP - Planering

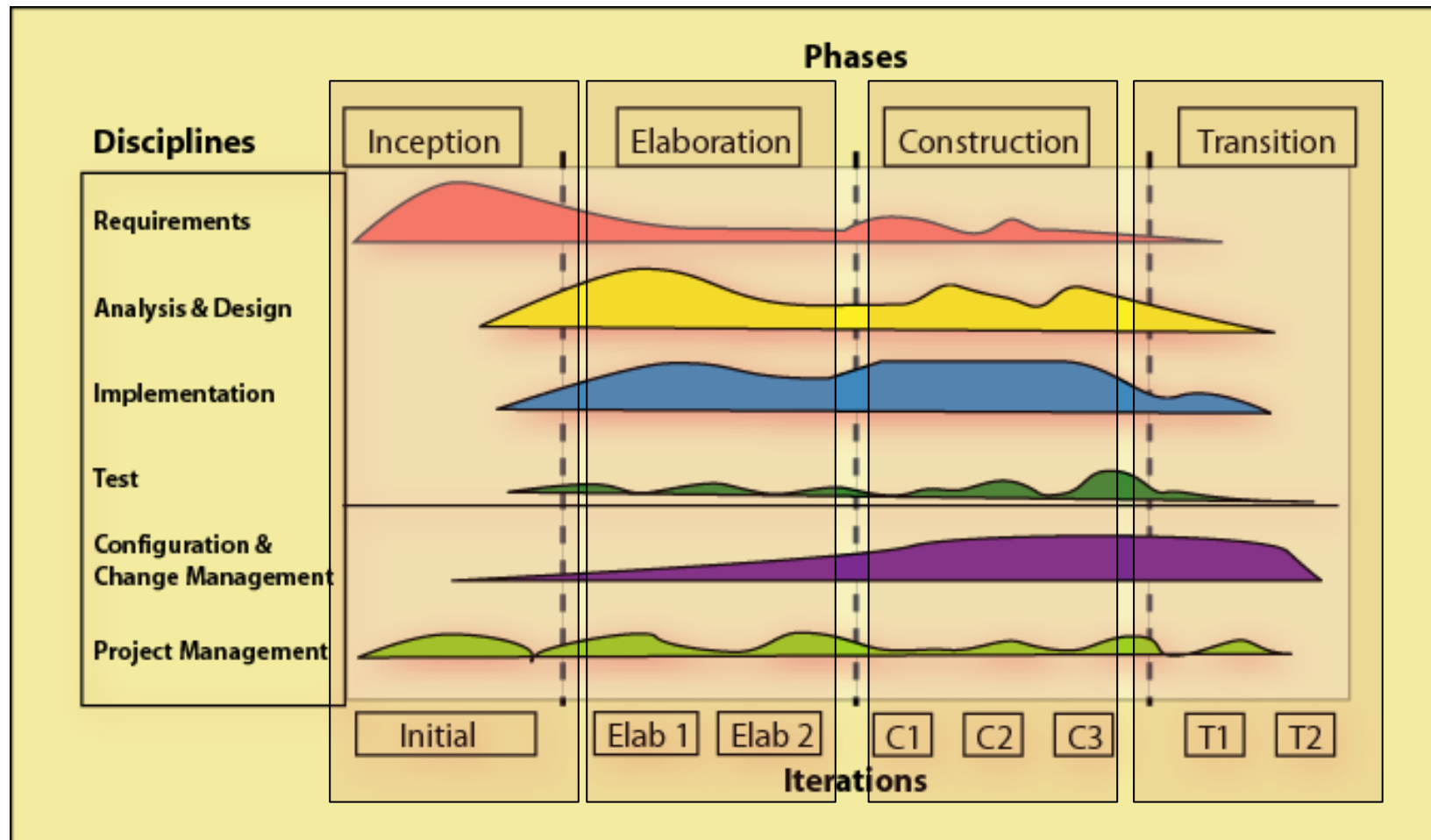
- ✓ **Grovkorning - Långsiktig**
 - Vision
 - Projektplan
 - Iterationsplanering
- ✓ **Finkorning - kortsiktigt**
 - Iterationsplan



Hur hanterar vi Iterationer och Inkrement – Open UP



UP – Faser



Inception

- ✓ Mål
 - Få igång projektet
 - Ta fram en mängd artefakter som man kan arbeta vidare med.
- ✓ Fokus
 - Kund
 - Projekt
 - Risker
- ✓ Milstolpar
 - Definiera viktiga systemfunktioner
 - Definiera en kandidat mjukvaruarkitektur
 - Etablera projektomgivningen

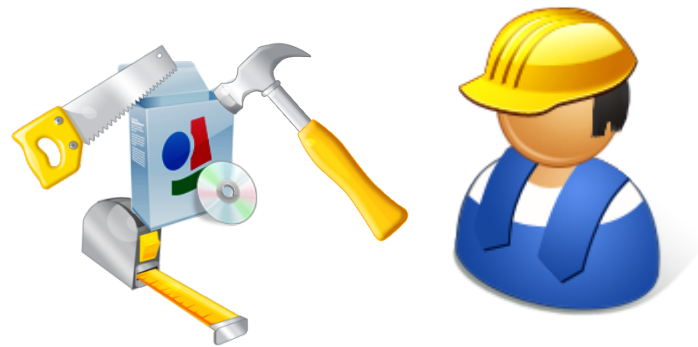


Elaboration

- ✓ Mål
 - Konstruera en fungerande körbar arkitekturbas
 - En detaljerad konstruktionsplan
- ✓ Fokus
 - Analys av kraven
 - Implementation av en arkitekturbas
- ✓ Milstolpar
 - En körbar arkitekturbas

Construction

- ✓ Mål
 - Utveckla arkitekturbasen till ett fullständigt mjukvarusystem
- ✓ Fokus
 - Implementation
 - Testning av komponenter
- ✓ Milstolpar
 - Fruset mjukvarusystem



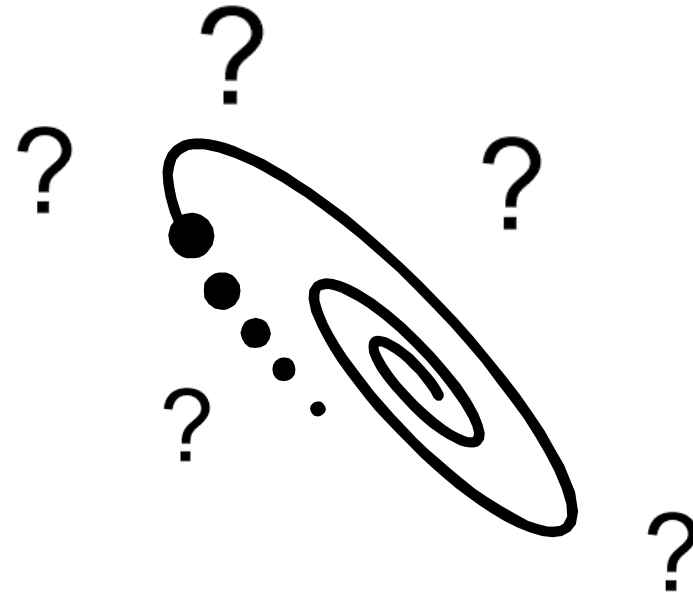
Transition

- ✓ Mål
 - Rätta fel/defekter
 - Genomför kundspecifika
- ✓ Fokus
 - Underhåll av design och implementation
 - Acceptans & Beta tester
- ✓ Milstolpar
 - RELEASE PARTY!



Inception

- ✓ Iterativt och Inkrementellt – vi måste börja någon gång
- ✓ Skapa en gemensam vision
- ✓ Avgränsa projektet
- ✓ Fokusera på
 - Var finns affären?
 - Grundläggande funktionella krav
 - Kritiska kvalitetskrav

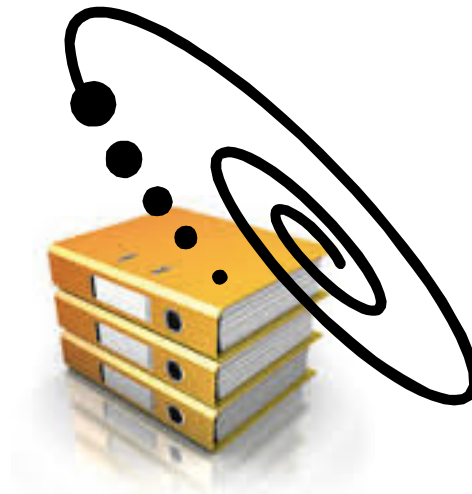


Vad är viktigt under inception-fasen

1. Förstå vad som skall konstrueras.
 - a. Skapa en övergripande vision, inklusive systemets gränser.
 - b. Identifiera intressenter Vilka är deras framgångskriterier?
2. Identifiera systemets nyckelfunktionalitet. Bestäm vilka krav som är mest kritiska.
3. Identifiera minst en möjlig lösning.
 - a. Bedöm om visionen är tekniskt möjligt.
 - b. Detta kan innebära att identifiera en kandidat till högnivåarkitektur eller skapa prototyper.
4. Gör en uppskattning på en hög nivå för
 - a. kostnader,
 - b. planer,
 - c. Risker

Artefakter under Inception-fasen

- ✓ Visionsdokument, mål och avgränsningar (executive summary)
- ✓ Use-case modell
- ✓ Supplementary Specification
- ✓ Risker och Riskhanteringsstrategier
- ✓ Prototyper
- ✓ Projektplan
- ✓ Iterationsplaner



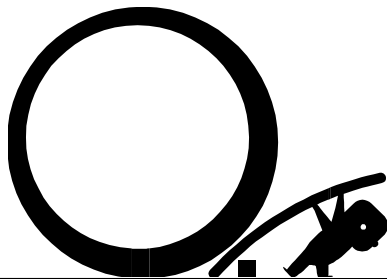
Mjukvaruutveckling

Problemlösning

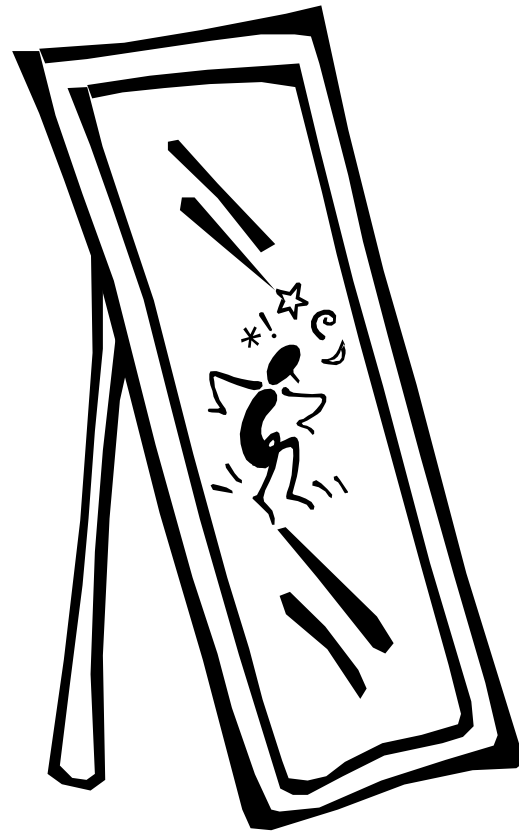
- ✓ Förstå problemet
- ✓ Tag fram en ide till lösning
- ✓ Genomför denna
- ✓ Utvärdera

Mjukvaruutveckling

- ✓ Krav
- ✓ Design
- ✓ Implementation
- ✓ Testning



Krav



Interna



Externa

Kravhantering

1. Samla in information
 - a. Användares önskemål.
 - b. Annan dokumentation.
2. Analysera och identifiera krav
 - a. Funktionalitet
 - b. Kvalitet
 - c. Process
3. Prioritera kraven
4. Specificera kraven på den produkt som skall tas fram.



Kom ihåg!

- ✓ Krav skall vara
 - Korrekta
 - Entydiga
 - Fullständiga
 - Verifierbara
 - Konsekventa
 - Designoberoende
 - Spårbara

Är detta möjligt?!?



Kravhantering är svårt

- ✓ Människor
- ✓ Kommunikation
 - Olika kulturer
 - Olika erfarenheter
- ✓ Tyst kunskap
- ✓ Olika agendor



Hur hanterar vi detta?

Varför bygger vi modeller av kraven?

- ✓ Fånga och kommunicera krav
- ✓ Förstå problemet och varandra!
- ✓ Överbrygga användarperspektivet och utvecklarperspektivet



Skriva en kravspecifikation?

- ✓ Språket viktigt!
 - Systemet skall ...
 - Systemet bör ...
 - Systemet kan ...
- ✓ Gruppera krav
 - Sortera ut det som absolut måste vara med
 - Övrigt diskuteras och
 - Inkluderas, eller
 - läggs till ”att göra i framtiden listan”

Kvalitetskrav

- ✓ ”Systemet skall vara snabbt”
- ✓ ”Systemet skall vara säkert”
- ✓ Vad göra?
- ✓ En specifikation skall vara
 - Precis
 - Entydig
 - Verifierbar

Exempel – Skriva krav

- ✓ **Original:** Software will not be loaded from unknown sources onto the system without first having the software tested and approved.
- ✓ **Bättre:** 3.2.5.2 Software shall be loaded onto the operational system only after it has been tested and approved.
- ✓ **Ännu bättre:** 3.2.5.2 Software shall be loaded onto the operational system only after it has been tested IAW MIL-SPEC 3425 and approved by the CCB.

Exempel II – Skriva krav

- ✓ 3.4.6.3 The system shall prevent the processing of duplicate electronic files by checking a new SDATE record. An email message shall be sent.
- ✓ 3.4.6.3 The system shall:
 - a. Prevent processing of duplicate electronic files by checking the date and time of submission.
 - b. Send the following email message:
 - Request updated submission of date and time, if necessary, or
 - That the processing was successful when successful.

Exempel III – Skriva krav

- ✓ 3.2.8.6 When doing calculations, the software shall produce correct results.

Ta bort!

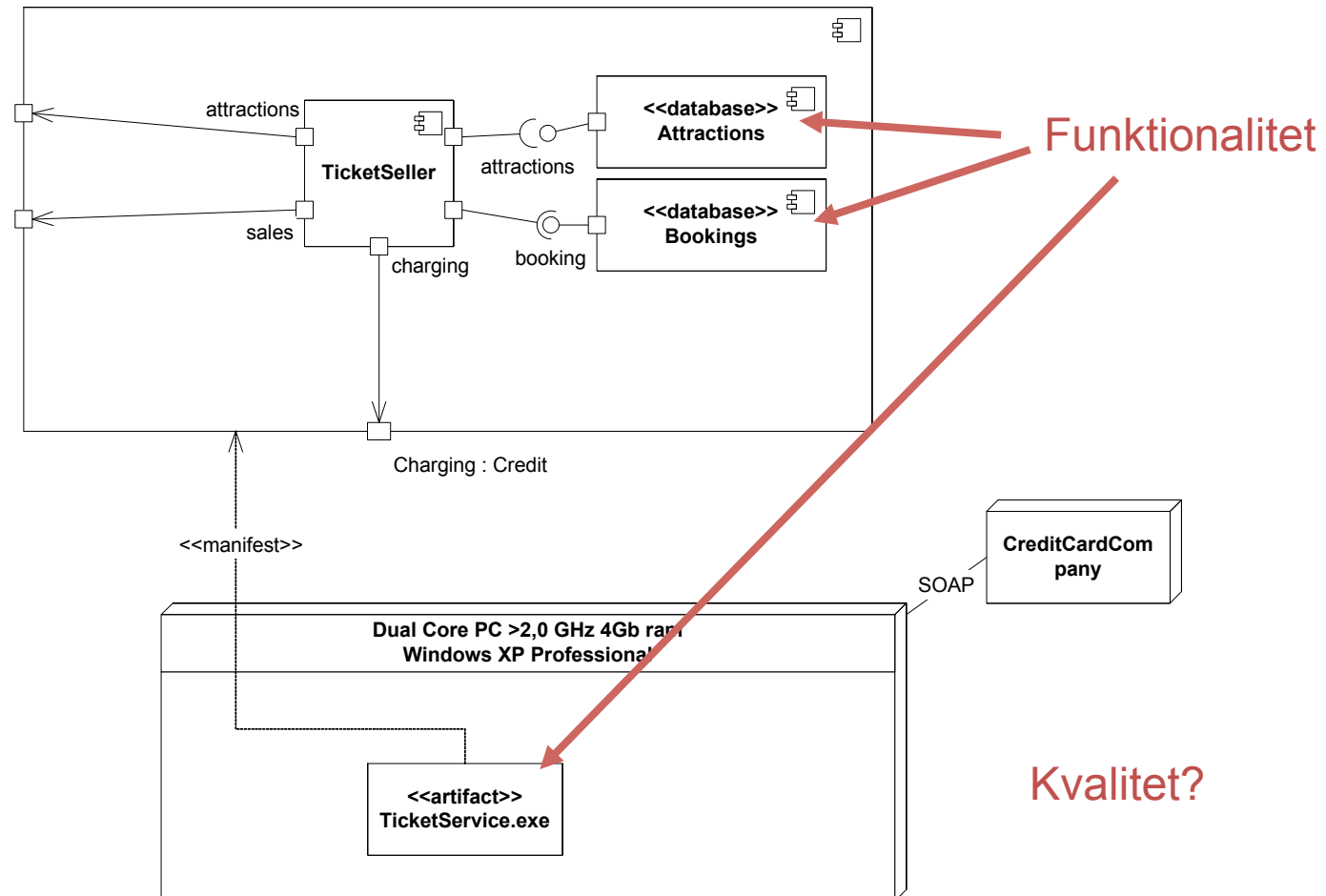


FURPS+ modellen (Grady 1992)

- ✓ FURPS – en checklista för krav.
 - **Functional** (features, capabilities, security)
 - **Usability** (human factors, help, documentation)
 - **Reliability** (frequency of failure, recoverability, predictability)
 - **Performance** (response time, throughput, accuracy, availability, resource usage)
 - **Supportability** (adaptability, maintainability, internationalization, configurability)

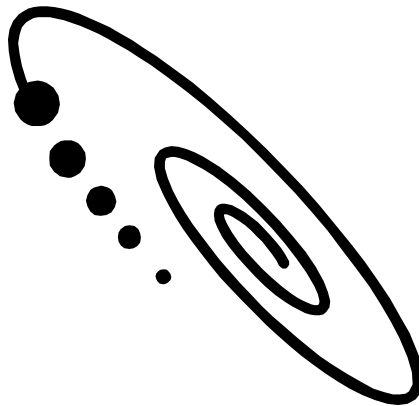


Krav



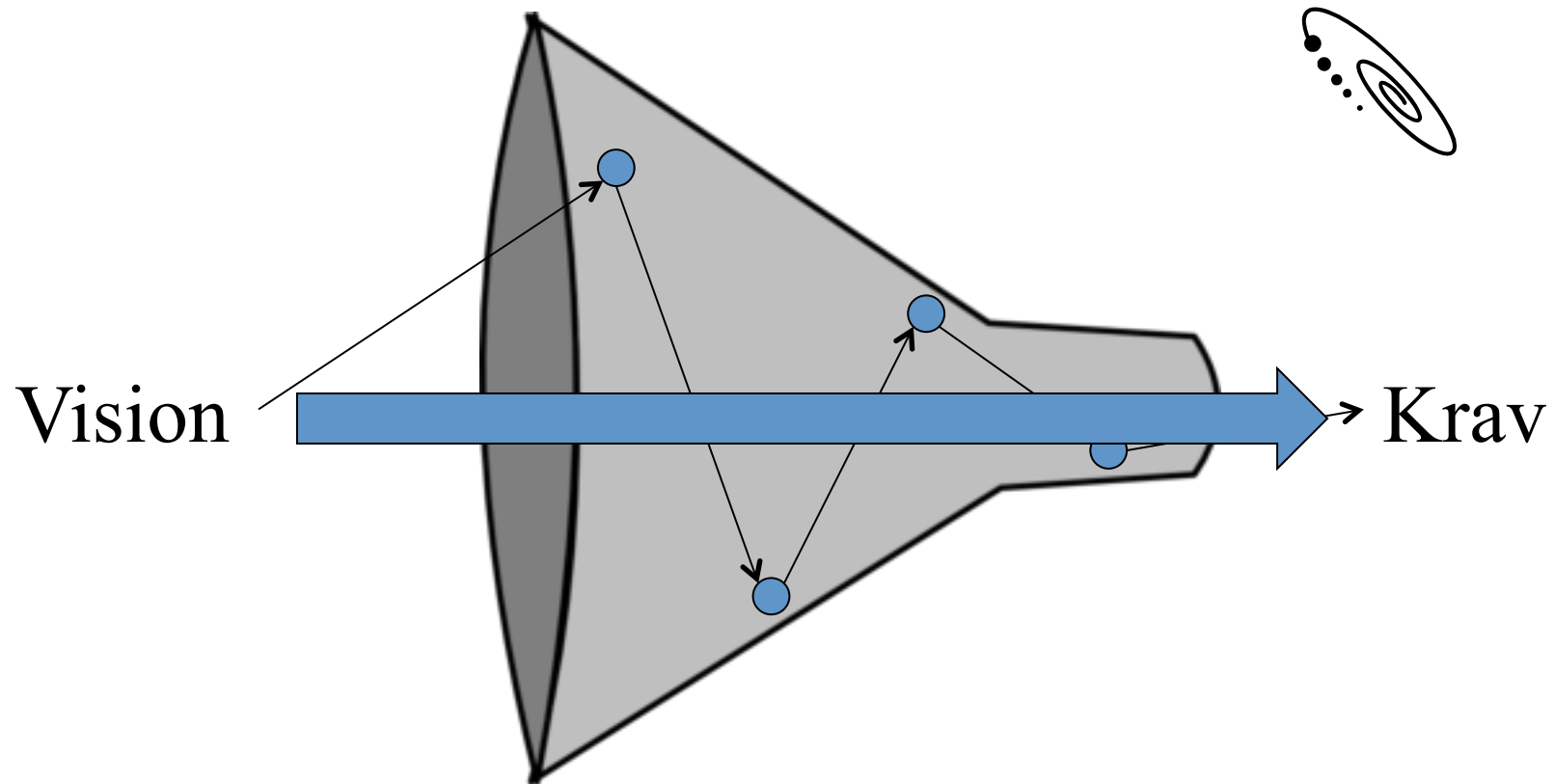
Krav måste växa fram

- ✓ Utnyttja iterationer och inkrement
- ✓ En stor del av problematiken beror på
 - Kommunikation!
 - Brist på förståelse mellan intressenter!
- ✓ Hur kan vi skapa förståelse mellan intressenter i ett utvecklingsprojekt?

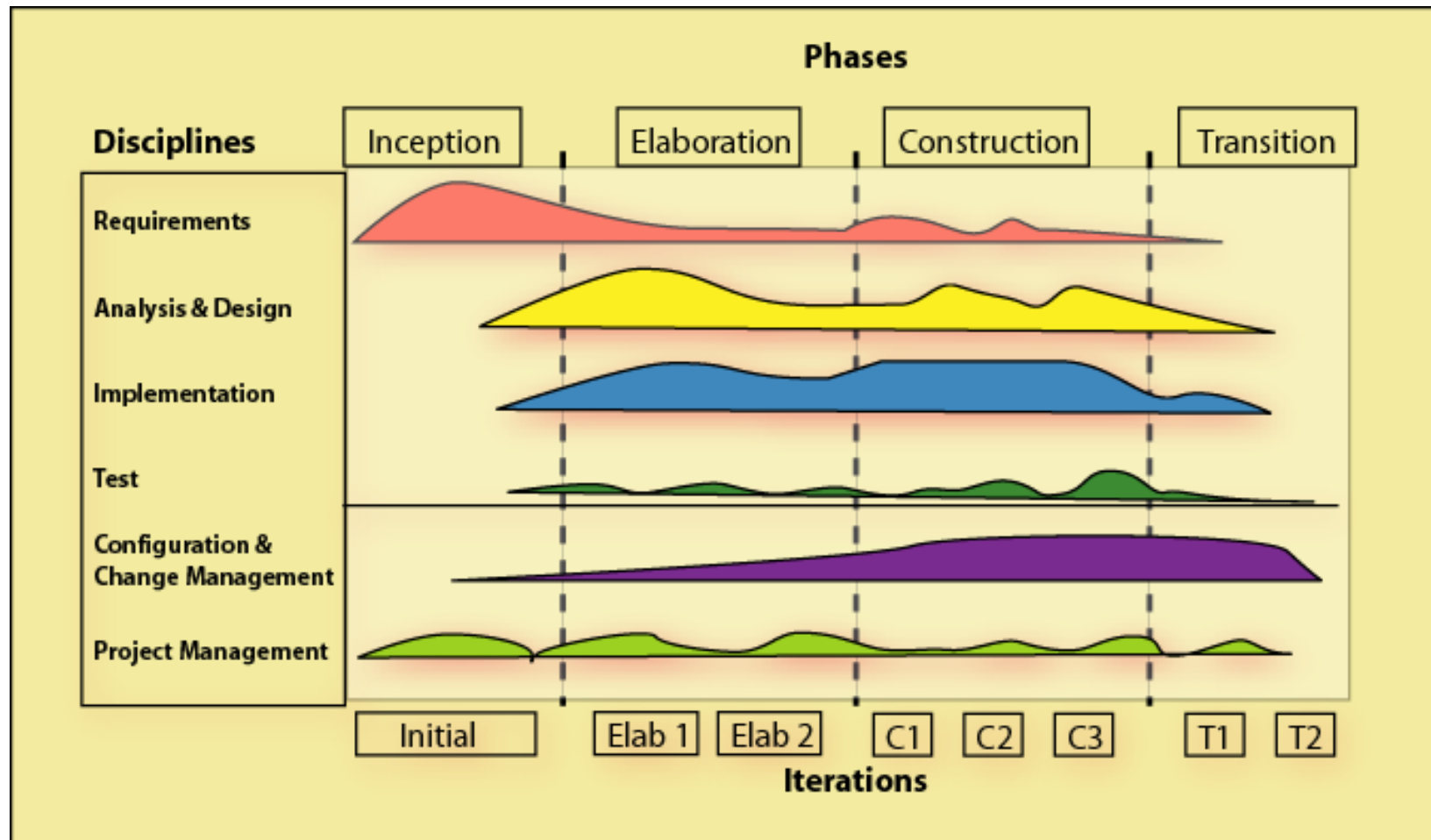


Krav växer fram

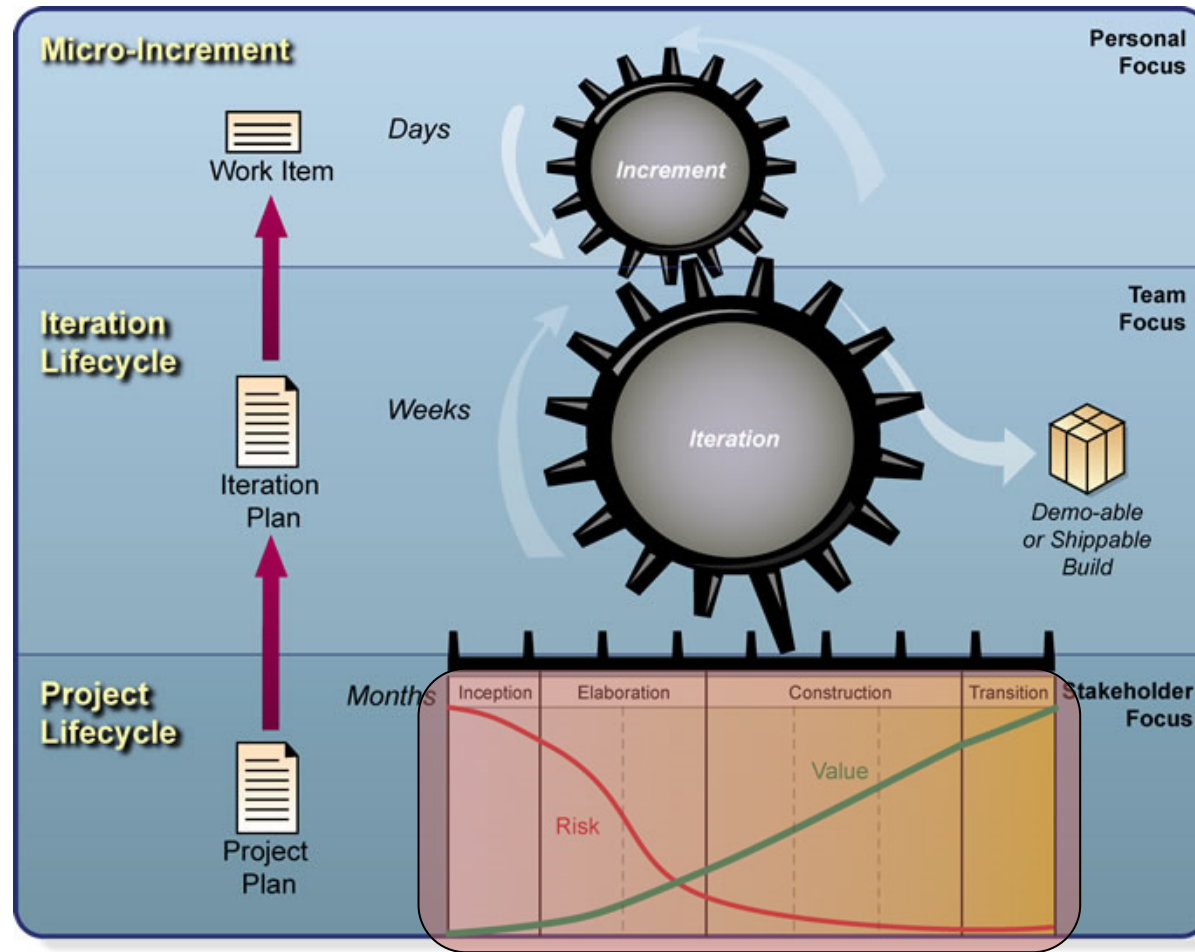
- ✓ Jämför **Iterativ framtagning** av krav med **“vattenfallsmodeller”**



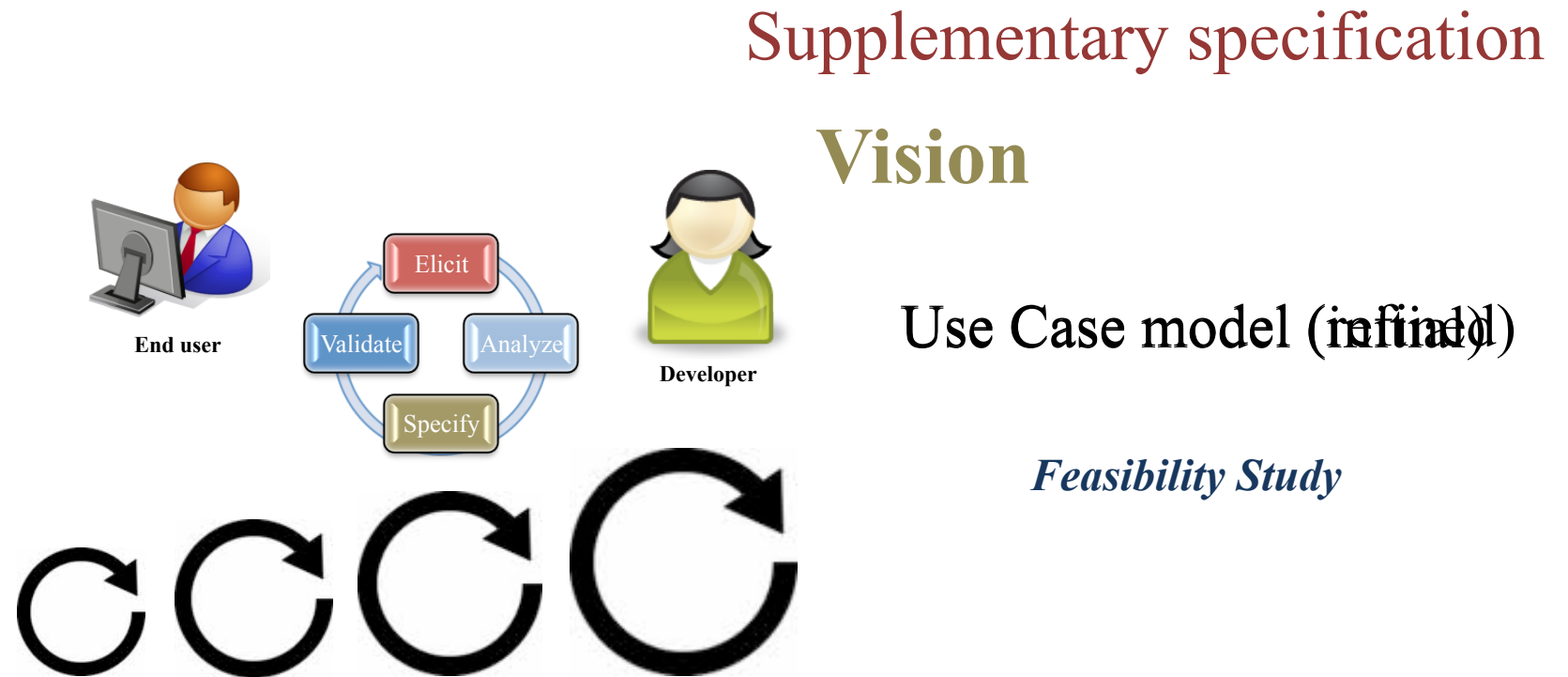
UP – Översikt



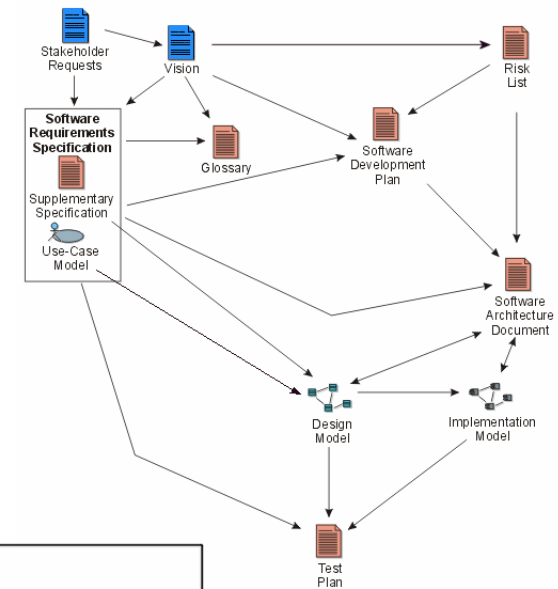
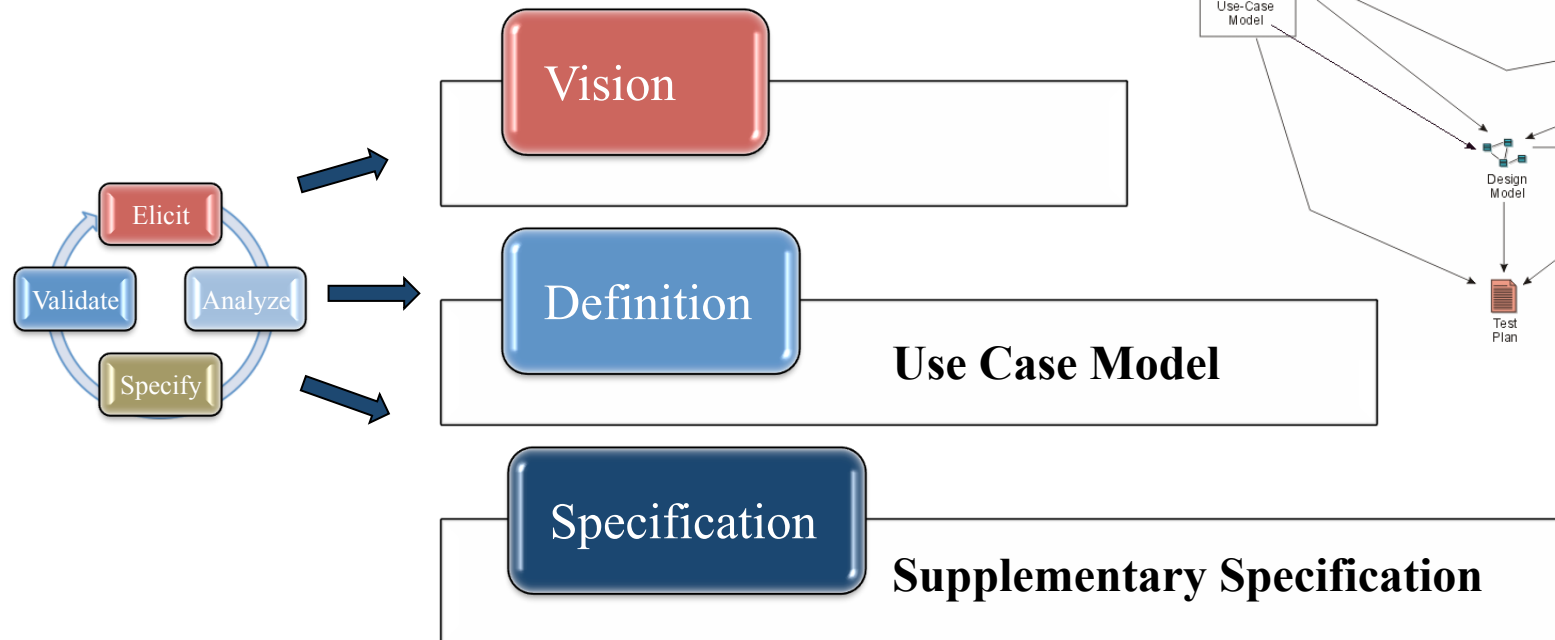
Open UP



Iterativ och Inkrementell



En Top-down ansats för kraven



Användningsfall

- ✓ Ett användningsfall berättar en historia om hur ett mål kan uppfyllas
- ✓ Eller en uppsättning av berättelser om både lyckas och misslyckas.
- ✓ Ett användningsfall utförs av *aktörer* under en begränsad tid
 - det finns en början
 - minst ett slut (multipel)
 - som bör nås inom rimlig tid.
- ✓ Undersöka Mål för ett system är en bra utgångspunkt för att identifiera krav.

- ✓ *Vilka tjänster erbjuder systemet*
- ✓ *Vilka användare finns till systemet*

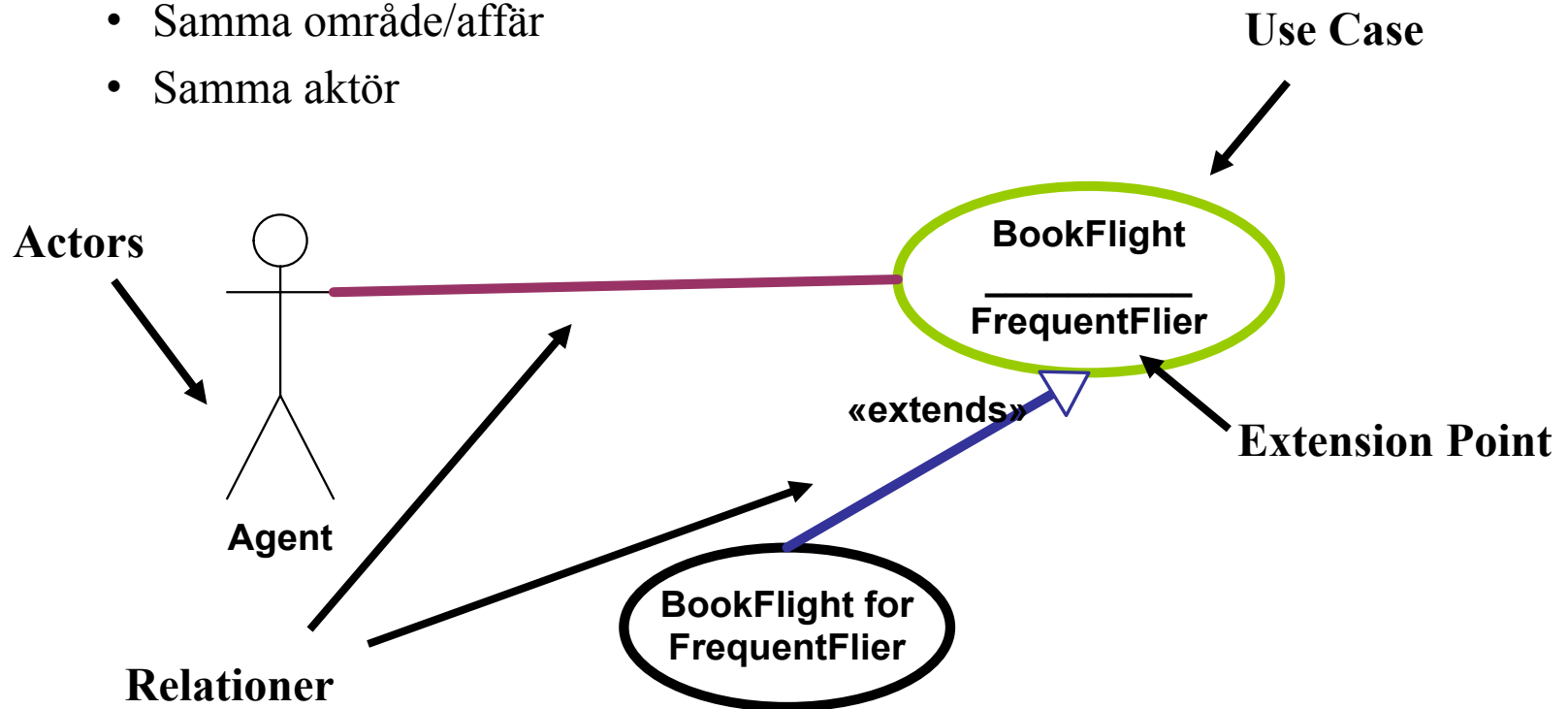
Use cases och Scenarier

- ✓ Scenarier är instanser av användningsfall
- ✓ Scenario – Use case liknar relationen mellan Objekt – Klass
- ✓ “Brainstorma” och dokumentera scenarier
 - Samla in detaljer
 - Kommunikation med användare
 - Perfekt start för att producera test-fall



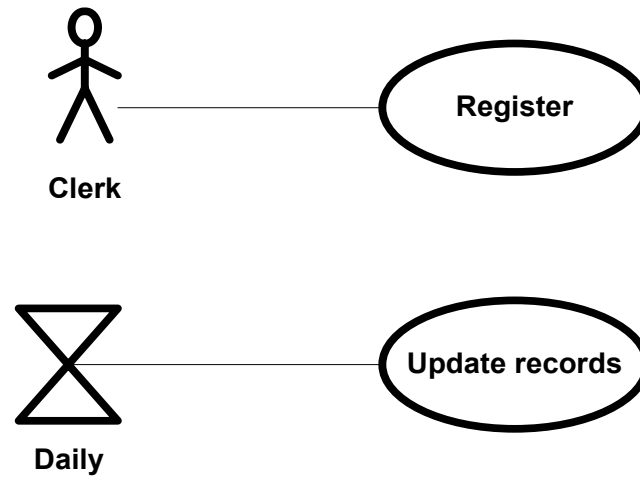
Use Case Modeller – Use Case Diagram

- ✓ Vi kan organisera användningsfall som modeller
 - Användningsfall grupperas efter sin logiska tillhörighet
 - Samma område/affär
 - Samma aktör



Actors

- ✓ **Klassiferar** s.k typanvändare
- ✓ En användare också vara kan var ett
 - System
 - En händelse
- ✓ En actor kan utöka en annan actor



Use Case Beskrivningar

- ✓ Detaljer i ett användningsfall
 - **Pre** och **post** villkor
 - **Primära** händelseflöden, ett sk. primärt flöde
 - **Alternativa** and **sekundära** flöden
 - Use case **relationer**



Exempel händelseflöde

BookFlight

1. Agent specifies a travel itinerary for customer
2. Searches a set of flights and displays them to the agent. (A)
3. Agent selects a flight.
4. Verifies space on selected flight (B)
5. Booking finalization, submitting payment information (C)
6. Books seats and issues tickets

Alternatives

- 4a. Space not available in selected category
 - 4a1. Informs agent
 - 4a2. Agent specifies another price range
- 4b. Seats not available (fully booked)
 - 4b1. Informs agent
 - 4b2. Agent specifies new departure preferences. (D)

