

Mjukvaruprocesser och Planering

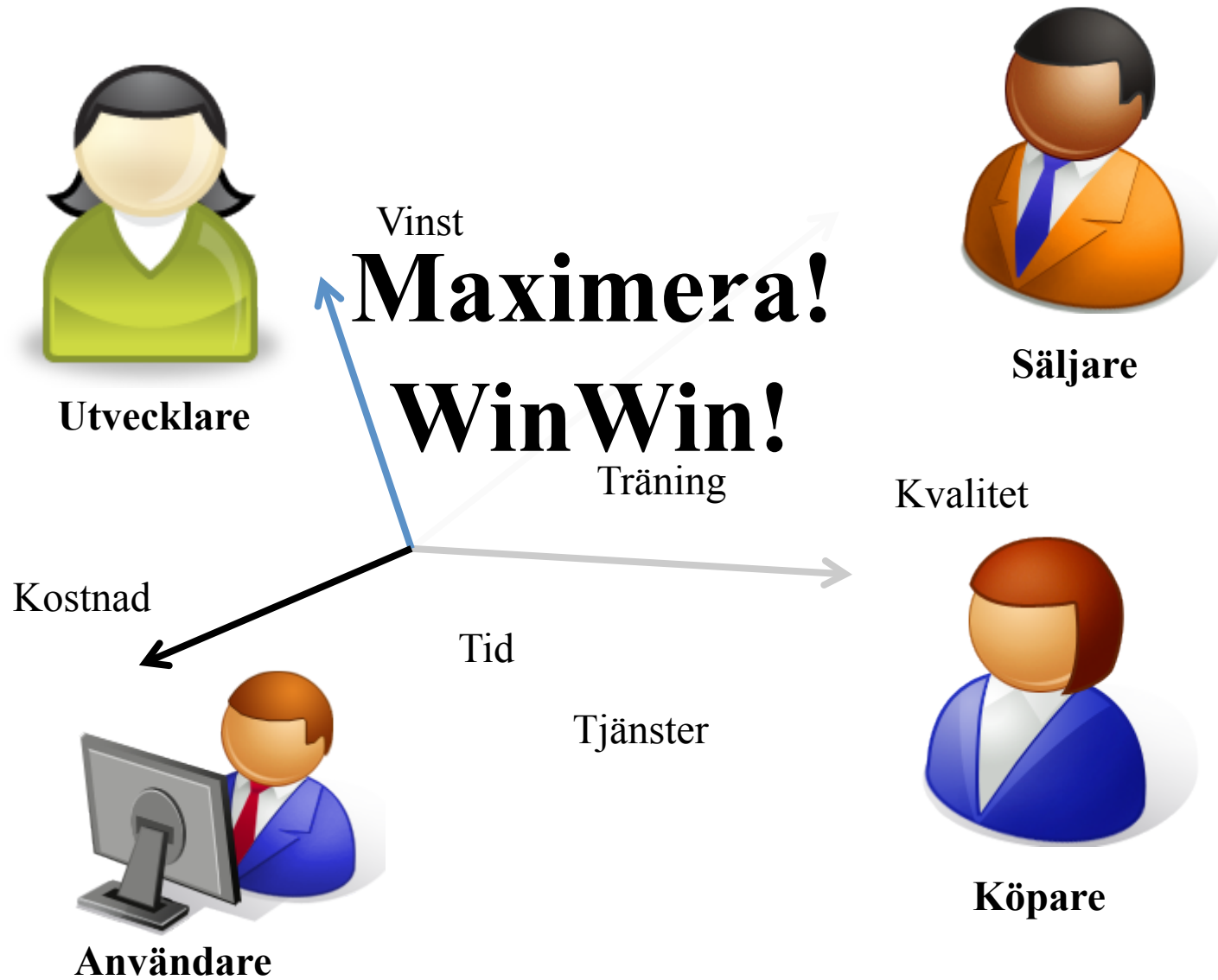
1DV404

HT14

Jesper Andersson



Mål



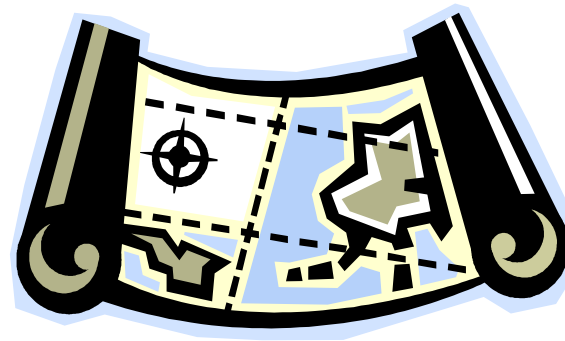
Och vi behöver en process?

- ✓ För att...
 - I ett företag måste alla resurser **arbeta mot samma mål**.
 - Arbetet måste vara möjligt att **kontrollera**.
 - Arbetet måste kunna **upprepas**.
 - Resultaten från olika projekt måste vara i nivå med varandra.
 - **Organisera arbetet**



Mjukvaruprocesser är mallar för att skapa en...

- ✓ vägbeskrivning för framgångsrika mjukvaruprojekt.
- ✓ Framgång kräver att vi besvarar följande frågor.
 - Vad skall göras?
 - När skall det utföras?
 - Vem skall göra det?
 - Hur skall det göras?
 - (Varför ingår tom ibland!)

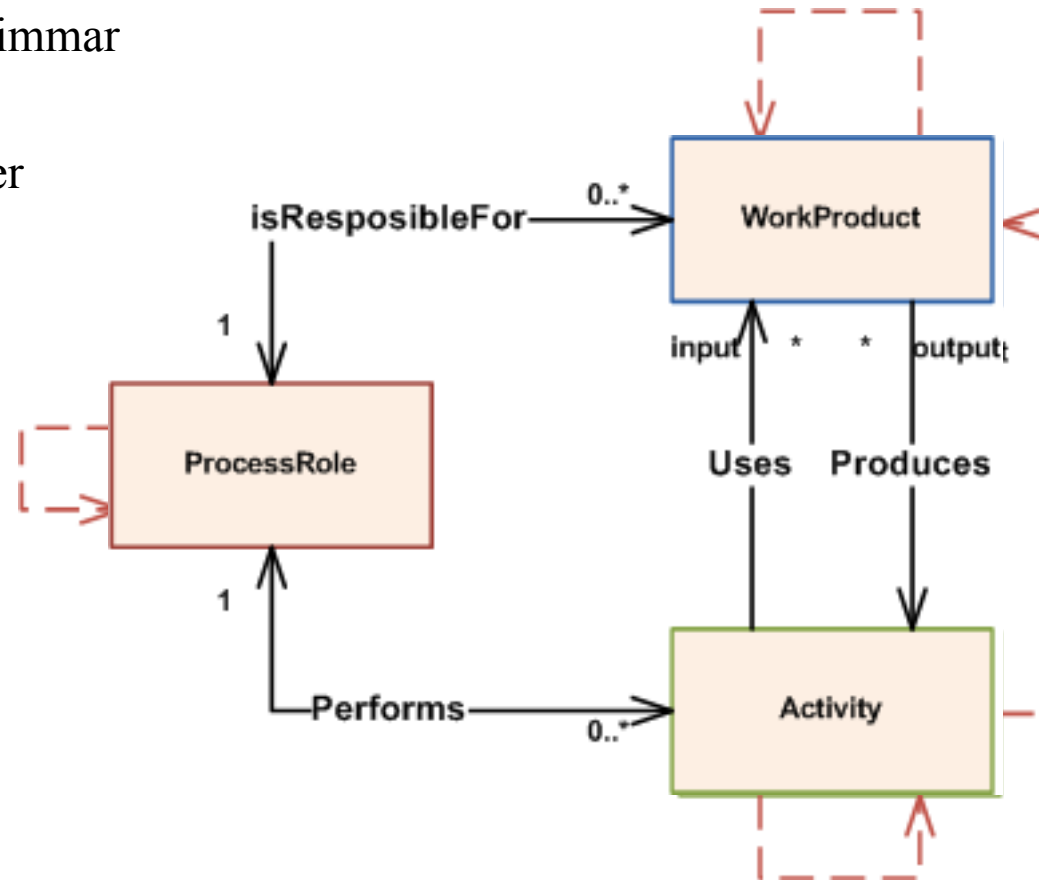


Hur organiserar vi arbetet?

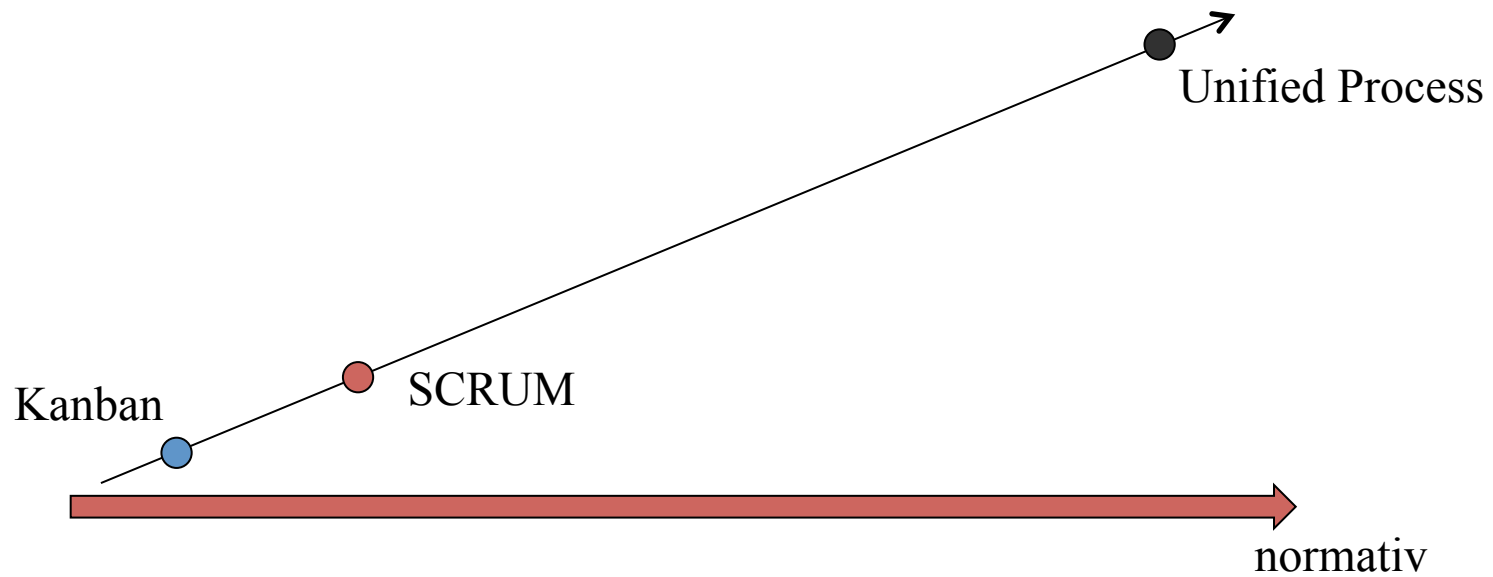
- ✓ *Roller* – Resurser – Mantimmar
- ✓ *Aktiviteter*
- ✓ *WorkProducts* – Artifakter

- ✓ Styrande faktorer
 - Tid
 - Resurser
 - Beroenden

- ✓ ➔ **Processmodell**



Processmodeller – Många smaker



Exempel Roller och några Artifakter i UP

Role Name	Project	Software Development	Deliverables	Requirements	Statistical	Issues	Risk	Glossary	Case Studies	Architecture	Candidates	Physical	Design	RUP	Work	Test	Test Plans	User Interface Design	Specification	Individual Name
Analyst Worker Set																				
Business-Process Analyst																				
Business Designer																				
Business-Model Reviewer																				
Requirements Reviewer										C						C			C	
System Analyst			C							C			C							
Use-Case Specifier										R										
User-Interface Designer																				R
Developer Worker Set																				
Architect		C					C			C			C	R			R		C	
Architect Reviewer		C												C						
Capsule Designer																				
Code Reviewer																				
Database Designer			C							C				R	C					
Design Reviewer																				
Designer			C											C	C					
Implementer			C							C				C	C					
Integrator			C											C	C					
Tester Worker Set																				
Test Designer																				
Tester			C							C							C			
Manager Worker Set																				
Change Control Manager																				
Configuration Manager																				
Deployment Manager																				
Process Engineer																				
Project Manager			R	R	R	R	R												C	
Project Reviewer			C							C										
Additional Worker Set																				
Any Worker																				
Course Developer																				
Graphic Artist																				
Stakeholder																				
System Administrator																				
Technical Writer																				
Tool Specialist																				



SCRUM Roller och Aktiviteter

✓ Roller

- Product Owner – ägare av produktvisionen
- Scrum Master – hjälper utvecklingsteamet att på bästa sätt använda SCRUM för att bygga produkten
- Development team – bygger produkten

✓ Artifakter

- Product increment – en färdigställd delmängd av en produkt.
- Product backlog – en prioriterad idélista för ett projekt
- Sprint backlog – en detaljerad plan för utvecklingen under nästa sprint.

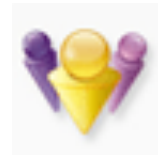


Aktiviteter i mjukvaruutvecklingsprojekt

- ✓ Kravhantering
- ✓ Kravanalys
- ✓ Design
- ✓ Implementation

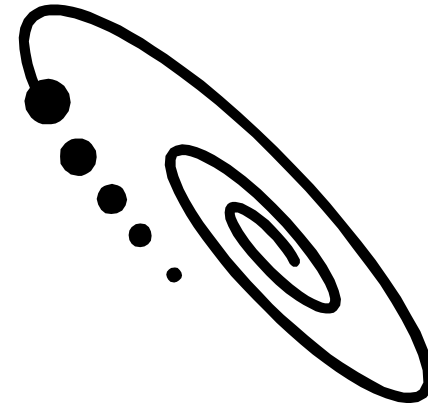


- ✓ Testning
- ✓ Integration
- ✓ Evolution

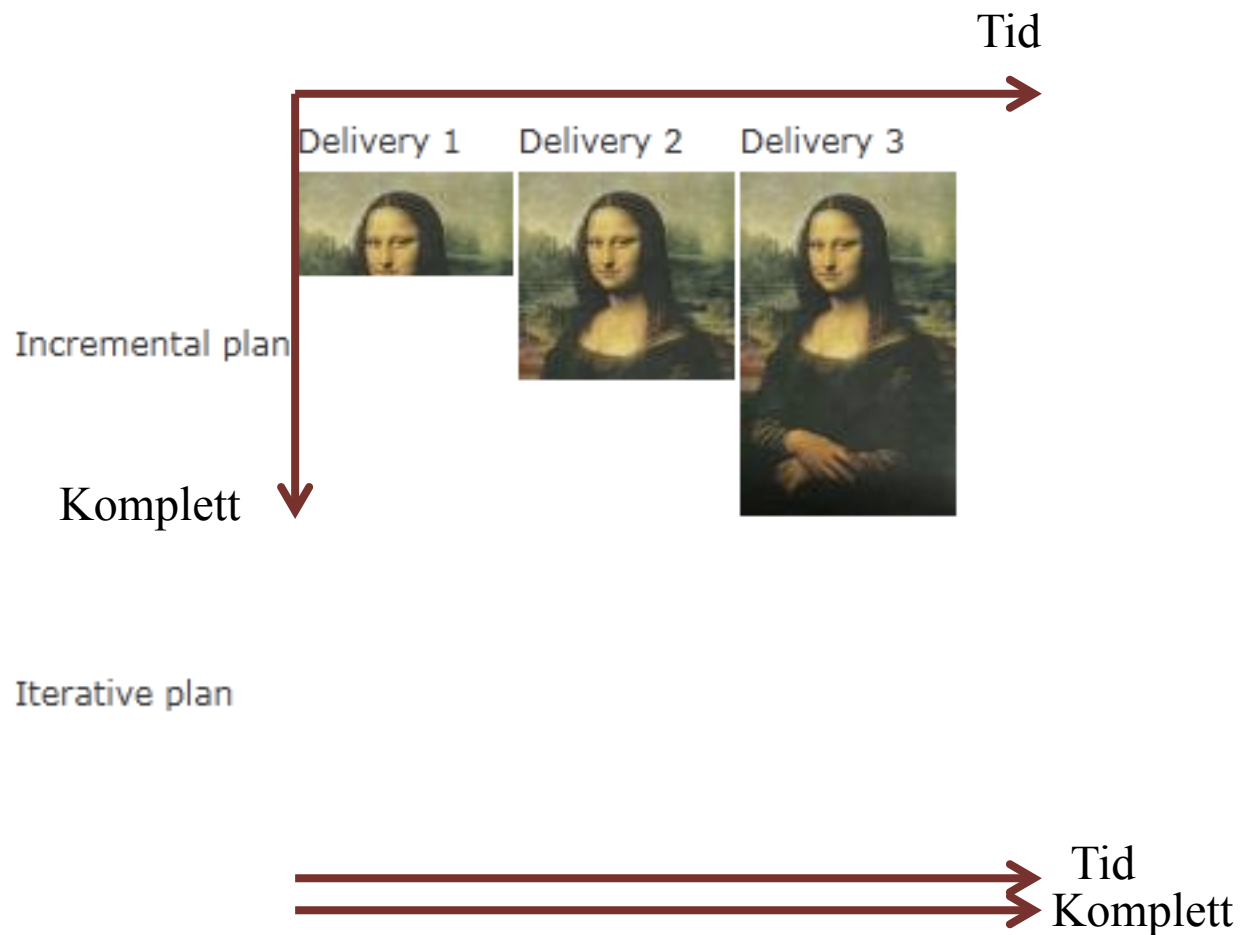


Iterationer & Inkrement

- ✓ Enklare att planera (!?)
 - Mindre steg
 - Kortare tid
 - Enklare att verifiera och validera
 - Snabbare återhämtning om vi gjort fel eller gått fel väg.



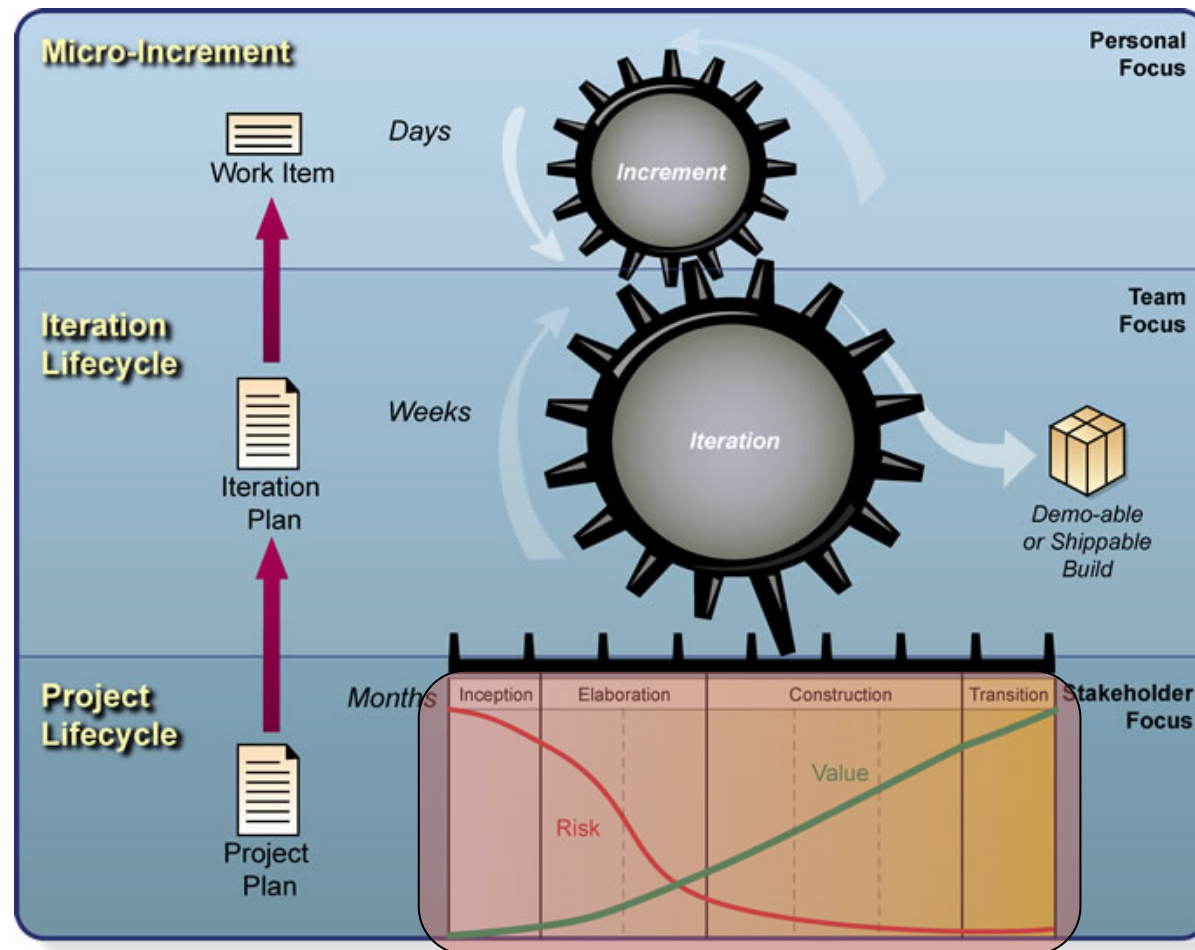
Iterativ vs. inkrementell ansats



Källa: <http://www.applitude.se/>

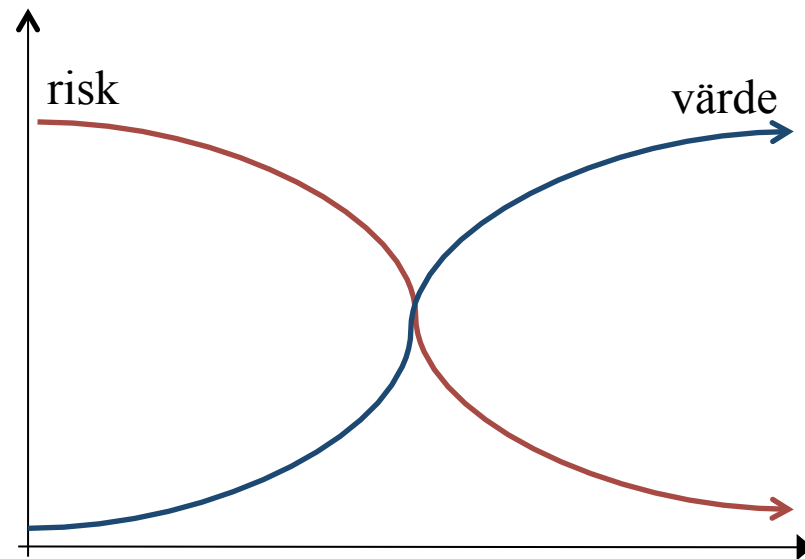


Hur hanterar vi Iterationer och Inkrement – Open UP



Riskhantering

- ✓ **Proaktiva** aktiviteter för att **minimera osäkerheter** och **eventuell skada** kopplat till ett utvecklingsprojekt.
- ✓ Beslut baserade på fakta och kunskap inte gissningslekar!
- ✓ Genomförs kontinuerligt under systemets livslängd! **Från dag 1!!**
- ✓ **Risker orsakas ofta av osäkerheter!**



Osäkerhet och osäkerhetstyper

<Osäkerhet> **orsakar** <Risk> **hanteras av** <åtgärd> **åstadkommer** <Resultat>

- ✓ *Komplexitet och Heterogenitet är primära faktorer för osäkerhet*
- ✓ Osäkerhetstyper
 - Brist på kunskap – *Fakta som inte är känd eller bara delvis känd och som måste finnas för att färdigställa produkten.*
 - Bristande definition – *Systemaspekter som inte specificerats eller beslutats.*
 - Statistiska variabler – *Oprecis kunskap, statistiskt säkerställd eller i alla fall avgränsad..*
 - Known Unknowns – *Saker som vi vet att vi inte vet.*
 - Unknown Unknowns – *‘Oooups’, saker vi inte vet.*



Osäkerhet: Risker/Möjligheter och Åtgärder

- ✓ Risker
 - Fel
 - Degradering.
 - Förändringar på marknaden.
 - Förändringar av behov.
- ✓ Åtgärder
 - Marginaler
 - Verifiering
 - Generalisering
 - Uppgraderingsbarhet



Risker and Risktyper

- ✓ **Teknik**
 - Databasen som används kan inte behandla så många transaktioner per sekund som förväntat.
 - Programvarukomponenter som ska återanvändas innehåller brister i funktionalitet.
- ✓ **Människor**
 - Det är omöjligt att rekrytera personal med den kompetens som krävs.
 - Nyckelpersonal är sjuk och otillgänglig vid kritiska tidpunkter.
 - Nödvändig utbildning för personalen finns inte tillgänglig.
- ✓ **Organisatoriska**
 - Omorganisationer har medfört att olika avdelningar är ansvariga för projektet.
 - Ekonomiska problem framtvingar minskningar av projektbudgeten.
- ✓ **Verktyg**
 - Den kod som genereras av ett CASE-verktyg är ineffektiv.
 - CASE-verktyg kan inte integreras.
- ✓ **Krav**
 - Det görs ändringar av krav som kräver större omarbetningar.
 - Kunderna förstår inte konsekvenserna av krav ändras.
- ✓ **Uppskattningar**
 - Den tid som krävs för att utveckla programvaran är underskattad.
 - Takten med vilken nya defekter upptäcks har underskattas.
 - Storleken av programvaran är underskattat.



“Risker” som driver projektet

- ✓ Projektrisker och produktrisker
- ✓ Risker är inte nödvändigtvis negativa.
 - Analysen driver utveckling
 - Ger alternativ
 - Och möjlighet till Trade-off analys
- ✓ Planera varje möjlig (kritisk) händelse i förväg.

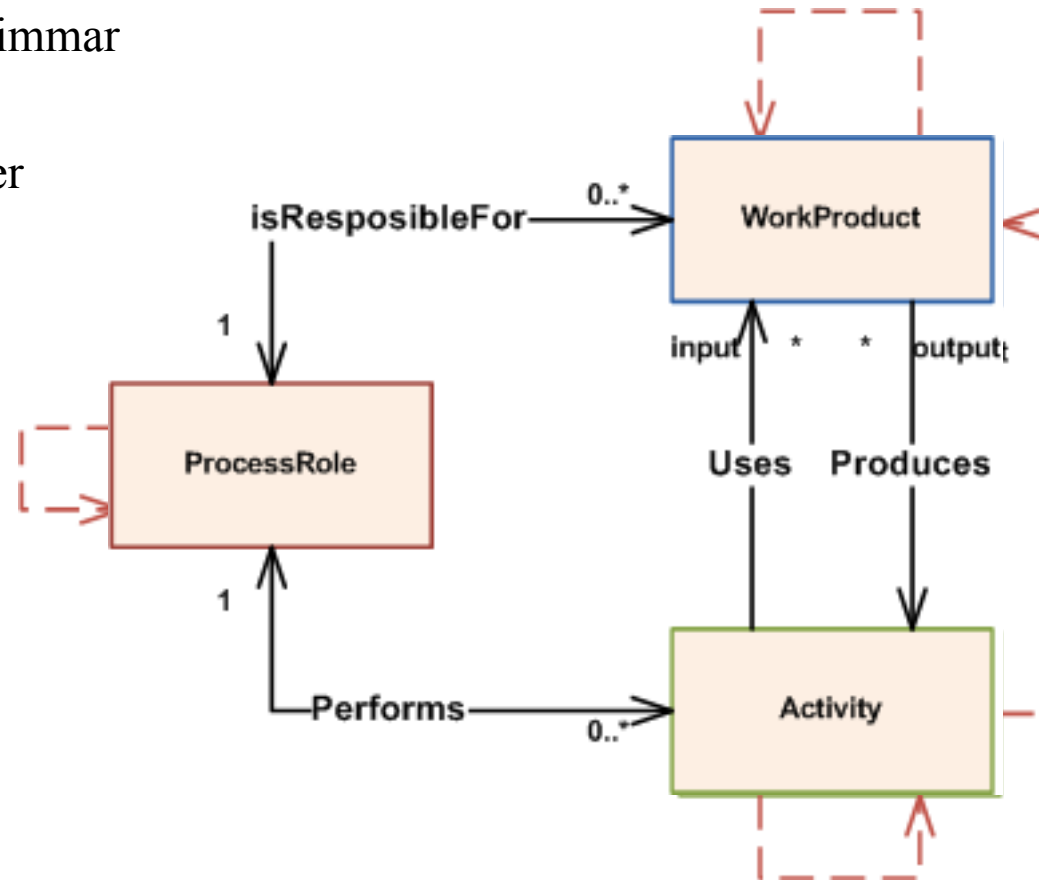


Hur organiserar vi arbetet?

- ✓ *Roller* – Resurser – Mantimmar
- ✓ *Aktiviteter*
- ✓ *WorkProducts* – Artifakter

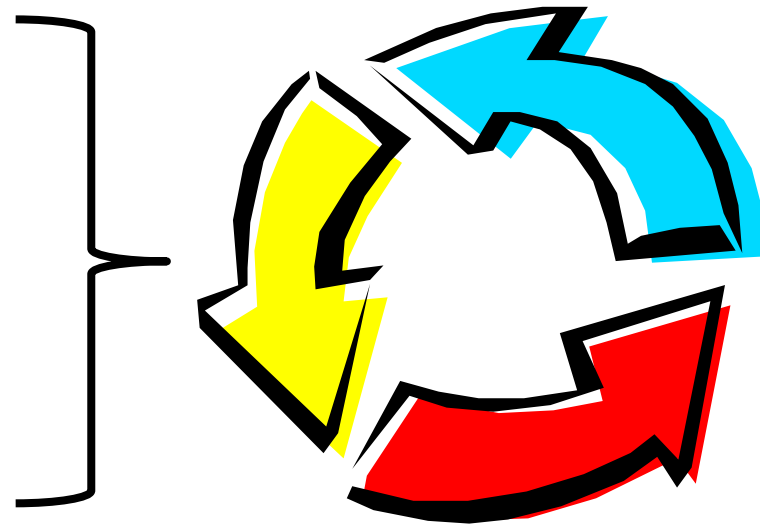
- ✓ Styrande faktorer
 - Tid
 - Resurser
 - Beroenden

- ✓ ➔ **Processmodell**



Några Processmodeller

- ✓ **Vattenfall (sekventiell)**
- ✓ **Evolutionary**
 - Upptäckande
 - Prototyping
- ✓ **Iterativa – Spiraler**
- ✓ **Agila**



Iterative

Vad är ett projekt?

- ✓ Ett projekt har ett mål som nås genom att slutföra uppgifter som är unika och inbördes beroende av varandra.
- ✓ Projekt genomförs genom att man använder och förbrukar resurser
- ✓ Projekten har
 - en omfattning,
 - scheman och
 - kostnader och
 - är genomförda inom vissa tidsramar,
 - budgetar, och
 - enligt specifikation



Mjukvaruprojekt, några kännetecken

- ✓ Stora projekt
- ✓ Distribuerade
- ✓ Osynlig produkt
- ✓ Alltid första gången ?!



Tecken på misslyckande (eller fara för)

- Projektledningen förstår inte användarnas behov.
- Projektets omfattning är dåligt definierad.
- Projektförändringar hanteras dåligt.
- Den valda tekniken förändras.
- Affärsidéen behöver förändras.
- Deadlines är orealistiska.
- Användarna är motsträviga.
- Stöd förloras.
- Projektet saknar folk med rätt kompetens.
- Chefer ignorerar 'best practices' och erfarenhet.

Reel, J.S., IEEE Software, May/June 1999



Fem viktiga framgångsfaktorer

- ✓ Kom igång direkt.
- ✓ Behåll drivkraften.
- ✓ Mät framstegen.
- ✓ Fatta smarta beslut.
- ✓ Genomför alltid “post-mortem” analyser.



Principer för planering

- ✓ Planering på kort sikt är lättare än att planera på lång sikt!
 - Okända risker
 - Resurser och mål
- ✓ “Gårdagens väder är den bästa prognosen”
- ✓ Planera med ”slack” - ett förändrat startdatum är ett förändrat slutdatum
- ✓ Mongoliska hjordens strategi - Antagandet att lägga till folk påskyndar genomförandet av en uppgift är inte sant!
- ✓ Matcha kompetens och erfarenhet med uppgiftens egenskaper
- ✓ Balansera grupperna



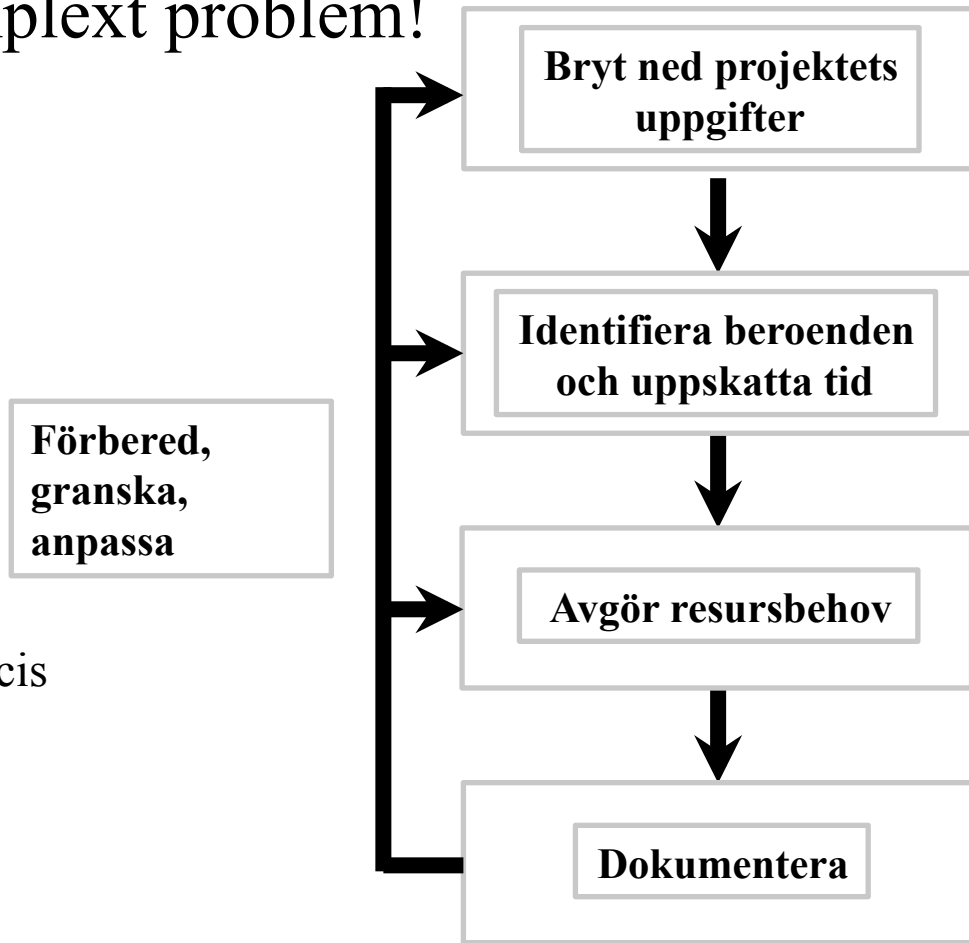
Planering

- ✓ Två strategier
 - Top-down, traditionell ansats
 - Bottom up, en agil ansats
- ✓ Top down
 - Använd en existerande processmodell och matcha projektet med den.
 - Instantiera processen!
 - Allokera resurser till processens aktiviteter.
 - Definiera projektspecifika milstolpar.
- ✓ Bottom up – Planeringsspelet
 - Planera 'releases'
 - Planera iterationer



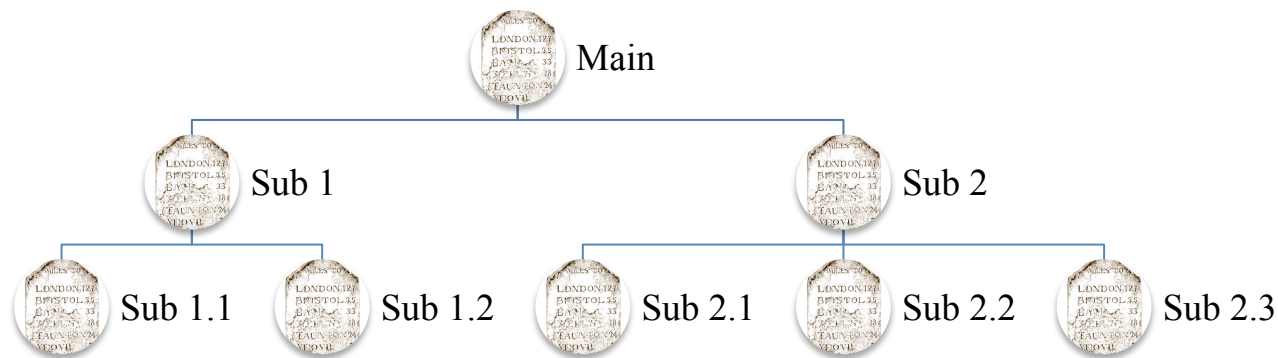
Planera – att lösa ett komplext problem!

- ✓ Divide and conquer
- ✓ Bryt ned närtid i i fler detaljer!
- ✓ Behåll framtiden mer oskarp/oprecis

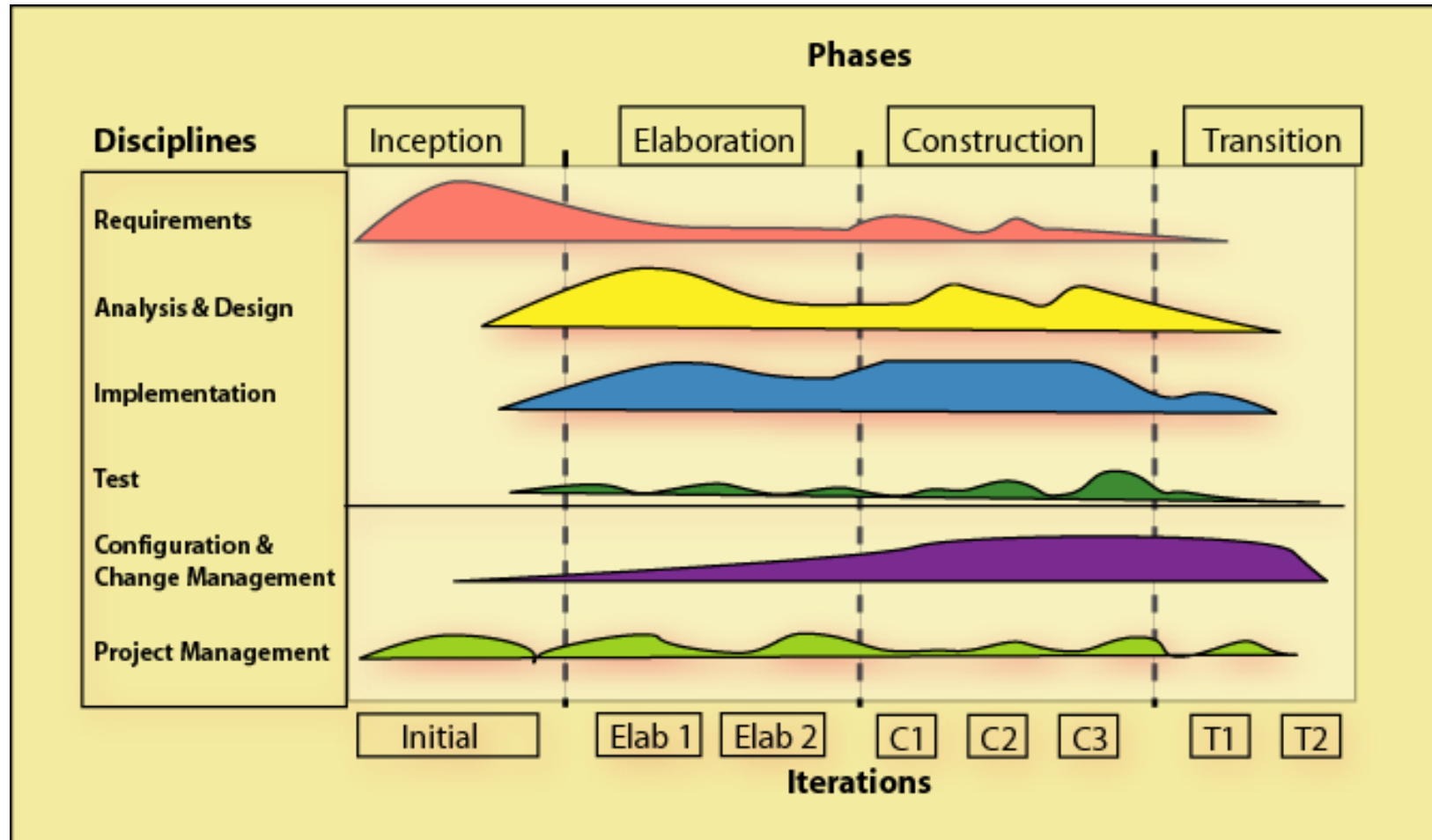


Work Breakdown Structures (WBS)

- ✓ En hierarkisk representation av aktiviteter och uppgifter
- ✓ Börjar med de viktigaste milstolparna
- ✓ Förfinar milstolpar till aktiviteter och uppgifter med egna milstolpar.
- ✓ Detta är en kontinuerlig process, en WBS är ett “aktivt dokument” som kommer att förändras under projektets gång.

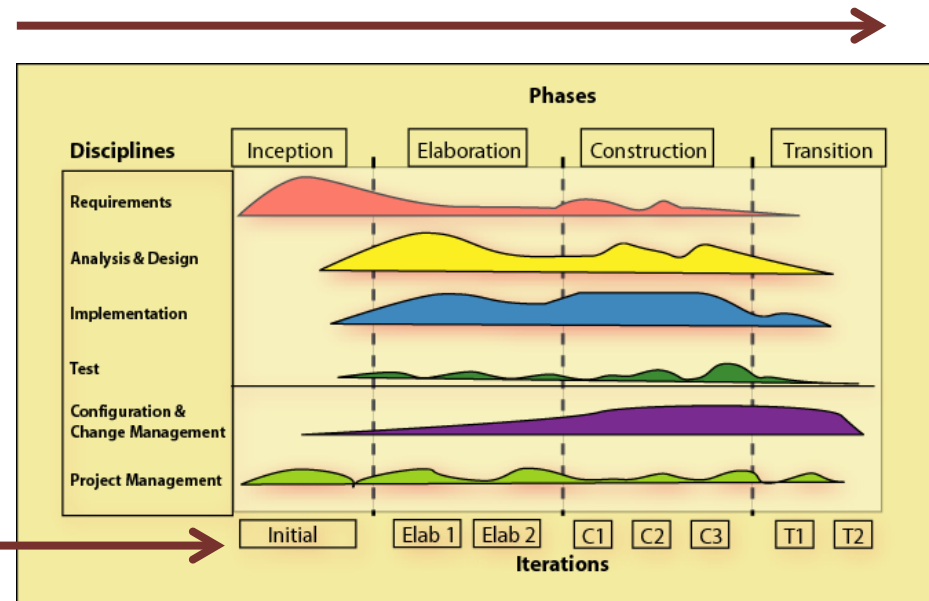


UP – Översikt



UP - Planering

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



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



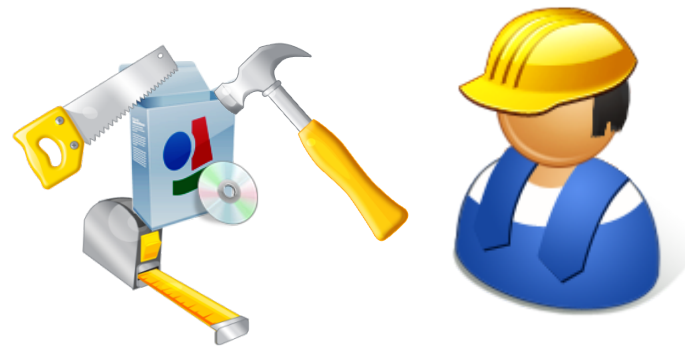
Elaboration

- ✓ 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

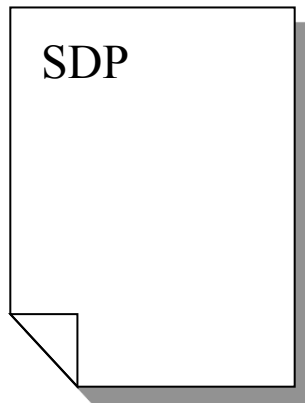


Transition

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

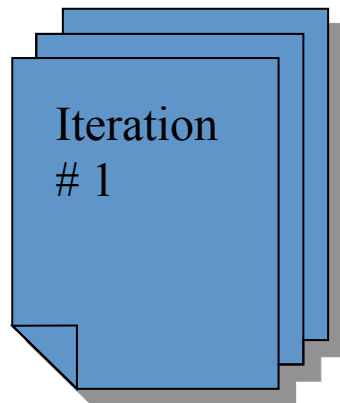


UP planering



- ▶ Fas-plan
- ▶ Iterationsplanering (per fas)

FAS



- ★ En *tidsordnad sekvens* av
 - ★ *aktiviteter*
 - ★ och *uppgifter*,
 - ★ med tilldelade *resurser*,
- ★ med *beroenden*,

Bottom-up (mer agilt)

- ✓ Agila principer
 - Självorganiserade grupper
 - Fungerande mjukvara istället för dokument
 - Kundsamarbete framför kontraktsförhandlingarna
 - Reaktion på förändringar istället för att följa en plan
- ✓ Två nivåer
 - Releaseplanering (med kunder)
 - Iterationsplanering (med utvecklare)



Releaseplanering

- ✓ Definiera milstolpar i form av “releaser”
- ✓ Del av att “definiera” projektets omfattning
- ✓ Genomförs tillsammans med kunden
- ✓ Förbered krav (användarberättelser) som ingår i kommande releaser (närvid)

- ✓ Faser
 - Explorativa fasen: Kunden ger en lista på krav med stort värde (användarberättelser).
 - Åtagandefasen: Bestäm åtagandet, dvs vilken funktionalitet som ska ingå i nästa version och releasedatum.
 - Genomförandefasen: Justera planen om det behövs. Till exempel kan krav sättas / ändras / tas bort



Iterationsplanering

- ✓ Planera utvecklarnas aktiviteter och uppgifter
- ✓ Allokera resurser till uppgifter
- ✓ Uppskatta resursåtgång.

- ✓ Faser
 - Explorativa fasen: ”Översätt” krav till uppgifter.
 - Åtagandefas: Uppskatta när uppgifterna skall vara klara och dela ut dem till utvecklarna.
 - Kontrollfas: Genomför uppgifterna och kontrollera resultatet mot det förväntade.



Ten Golden Rules of Project Management

1. Figure out what business you are in, and then mind your own business
2. Understand the customer's requirements and put them under version control.
3. Prepare a reasonable plan.
4. Build a good team with clear ownership.
5. Track project status and give it wide visibility.
6. Use Baseline Controls.
7. Write Important Stuff Down, Share it, and Save it.
8. If it hasn't been tested, it doesn't work.
9. Ensure Customer Satisfaction.
10. Be relentlessly pro-active.

