

Agil mjukvaruutveckling

1DV404,
Jesper Andersson

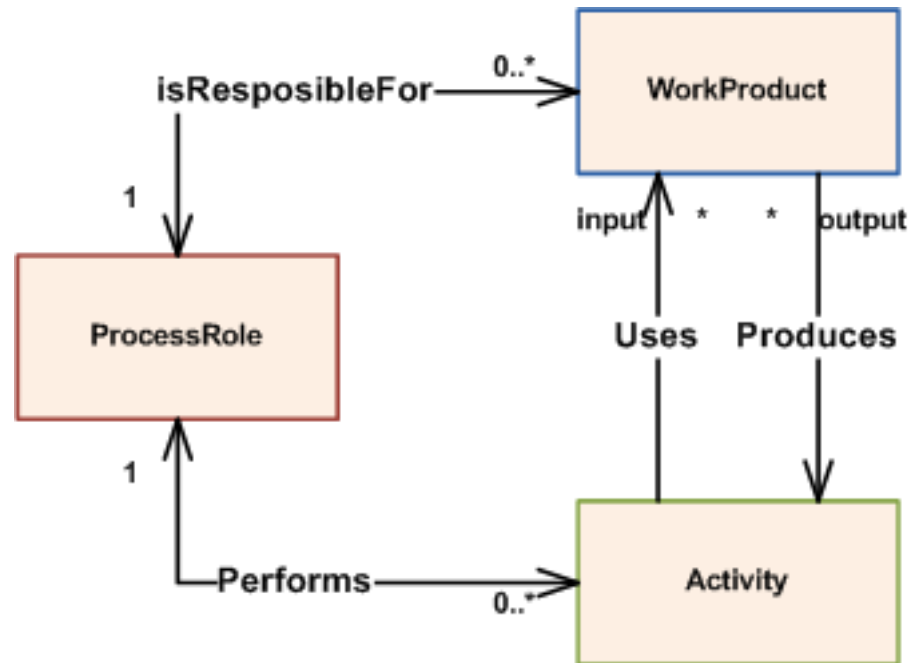


Agilt?

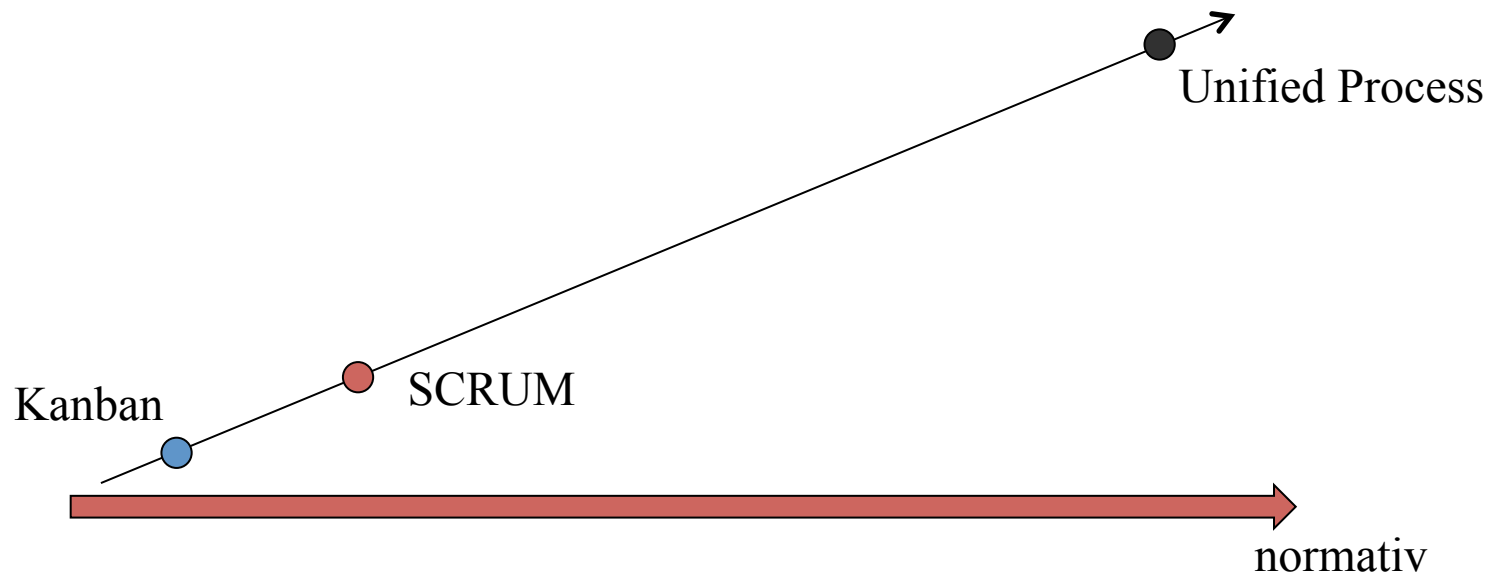


Innehållet i alla mjukvaruutvecklingsprocesser

- ✓ *Roller*
- ✓ *Aktiviteter*
- ✓ *Artefakter*



Processmodeller – Många smaker



Exempel Roller och några Artifakter i UP

Role Name	Proj ect Del iver able List	Soft ware Dev elop ment Plan ning	GAN T Chart	Stat us Rep ort	Issu e Log	Risk List	Glos sary	Cas e Stud ies	Can didat e Archit ect ure	Phy sical Model el	Desi gn Guid e	RUP Work shop es	Test Plans	User Inter face Desi gn Spe cific ation	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



Målet – Värdeskapande i focus

- ✓ Minimum Viable Product
- ✓ En produkt sammansatt av endast det mest nödvändiga funktionerna.
- ✓ Det som skapar mest värde till lägst risk.
- ✓ Lean development – minimera rest (waste)



Agila manifestet

- ✓ *Individer och interaktioner* framför *processer och verktyg*
- ✓ *Fungerande programvara* framför *omfattande dokumentation*
- ✓ *Kundsamarbete* framför *kontraktsförhandling*
- ✓ *Anpassning till förändring* framför *att följa en plan*

- ✓ Det finns värde i punkterna till höger men punkterna till vänster värdesätts högre



Principer bakom det agila manifestet

- ✓ Vår högsta prioritet är att tillfredsställa kunden genom tidig och kontinuerlig leverans av värdefull programvara.
- ✓ Välkomna förändrade krav, även sent under utvecklingen. Agila metoder utnyttjar förändring till kundens konkurrensfördel.
- ✓ Leverera fungerande programvara ofta, med ett par veckors till ett par månaders mellanrum, ju oftare desto bättre.
- ✓ Verksamhetskunniga och utvecklare måste arbeta tillsammans dagligen under hela projektet.
- ✓ Bygg projekt kring motiverade individer. Ge dem den miljö och det stöd de behöver, och lita på att de får jobbet gjort.
- ✓ Kommunikation ansikte mot ansikte är det bästa och effektivaste sättet att förmedla information, både till och inom utvecklingsteamet.

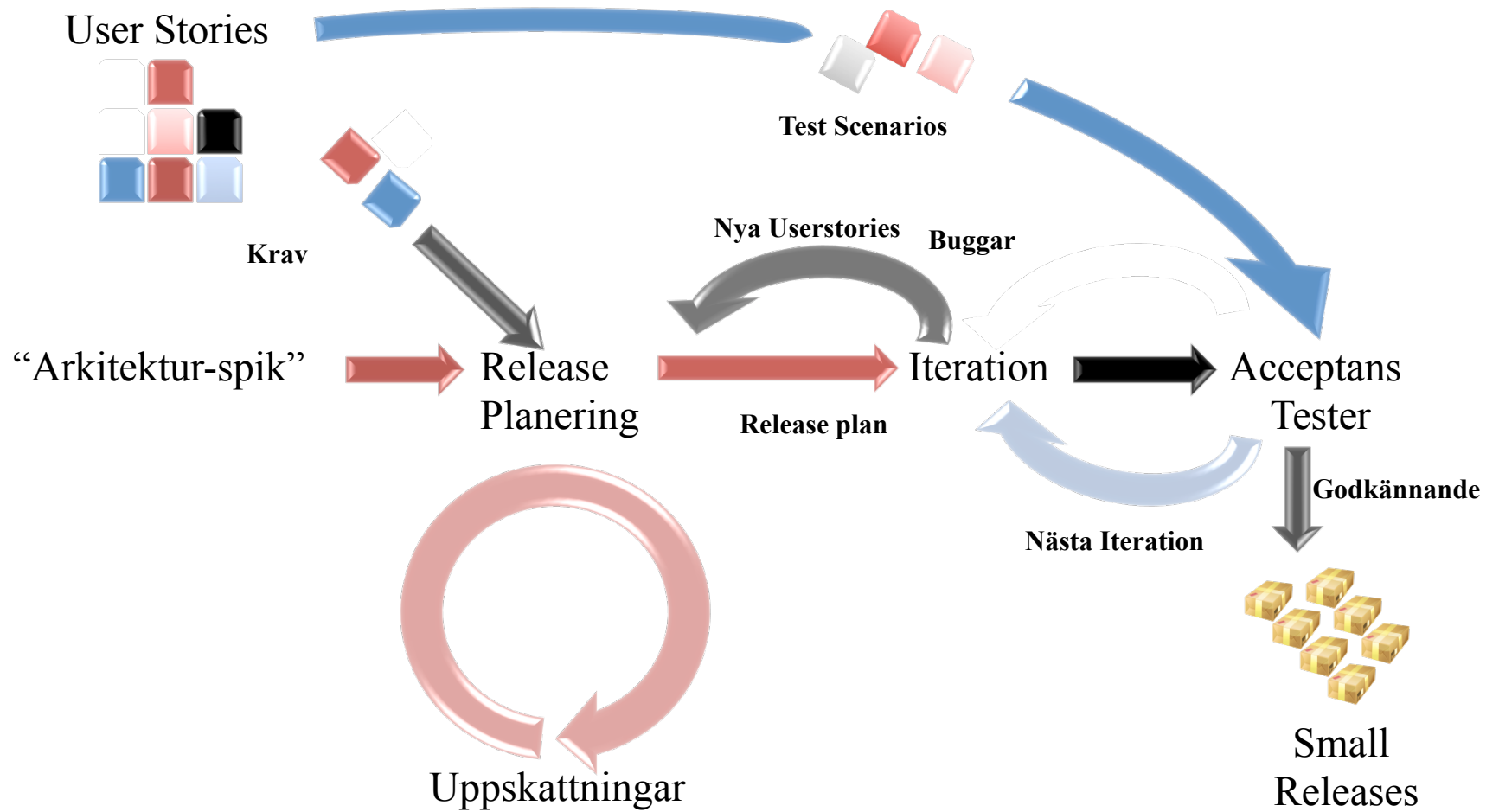


Principer bakom det agila manifestet

- ✓ Fungerande programvara är främsta måttet på framsteg.
- ✓ Agila metoder verkar för uthållighet. sponsorer, utvecklare och användare skall kunna hålla jämn utvecklingstakt under obegränsad tid.
- ✓ Kontinuerlig uppmärksamhet på förstklassig teknik och bra design stärker anpassningsförmågan.
- ✓ Enkelhet – konsten att maximera mängden arbete som inte görs – är grundläggande.
- ✓ Bäst arkitektur, krav och design växer fram med självorganiserande team.
- ✓ Med jämna mellanrum reflekterar teamet över hur det kan bli mer effektivt och justerar sitt beteende därefter.



XP – Första exemplet

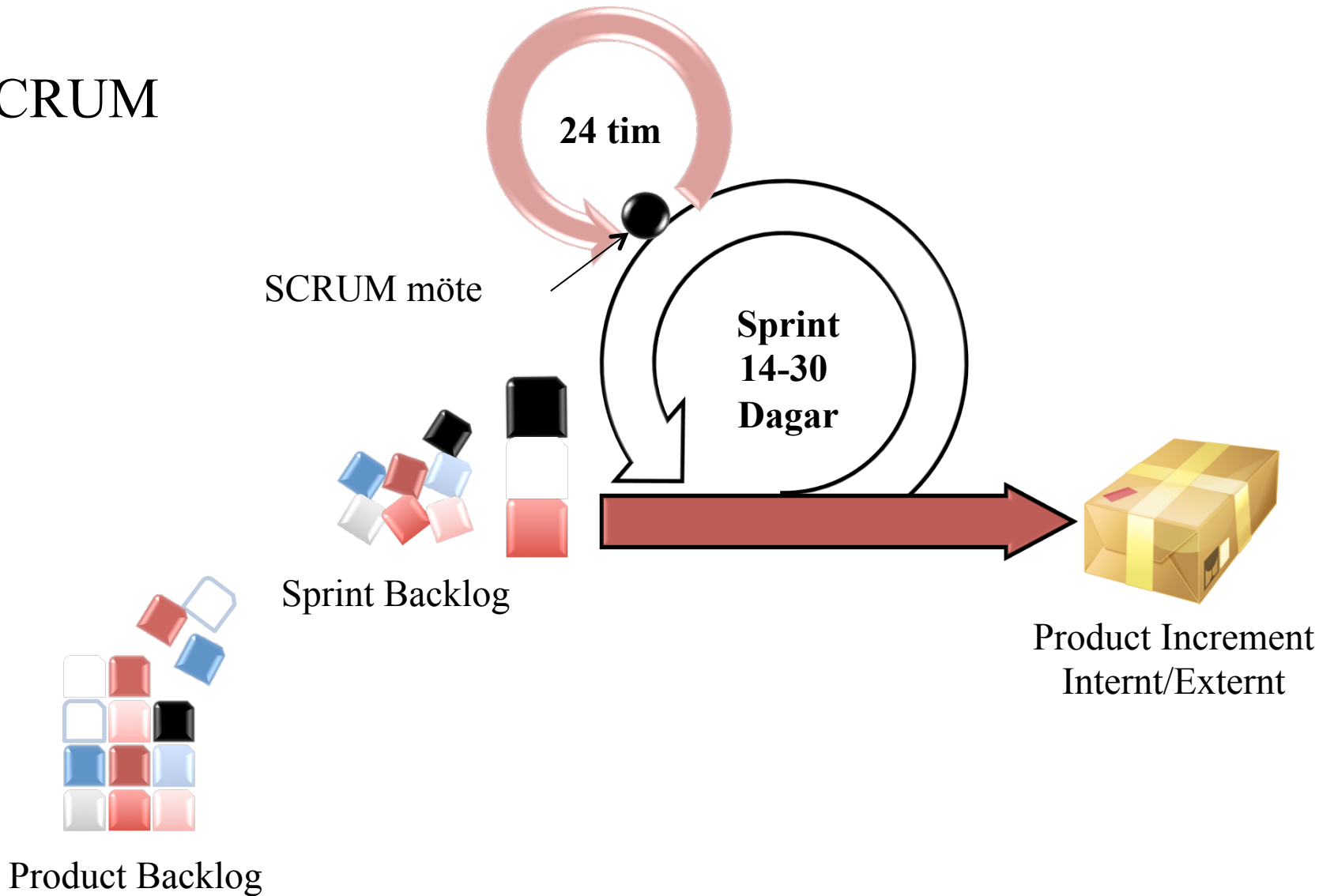


SCRUM

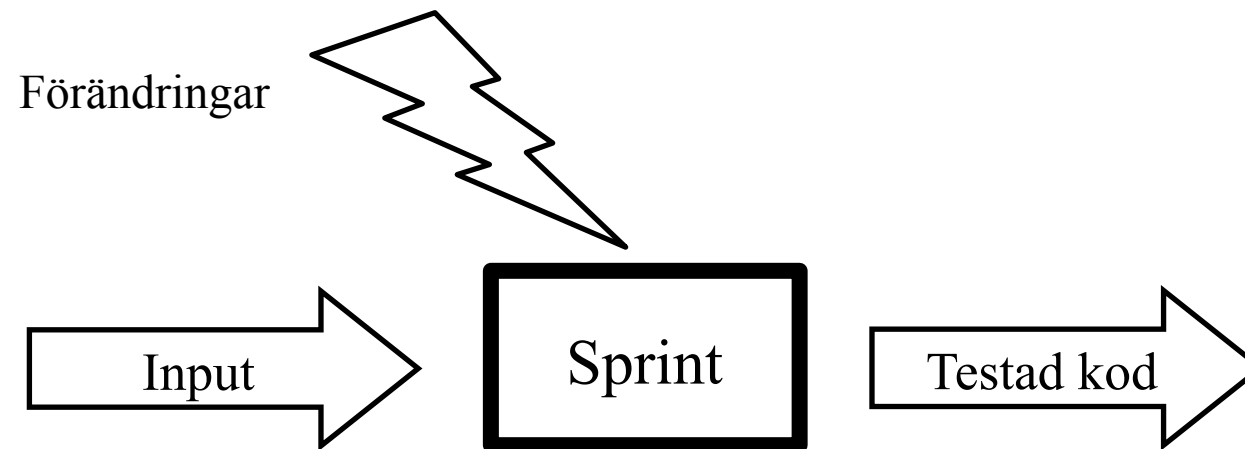
- ✓ En agil process som presenterades första gången 1996
- ✓ Viktigaste principen är “Self-organizing teams”, dvs delat ledarskap.
- ✓ Produkten växer fram i ett antal s.k. “sprints”.
- ✓ Kraven samlas i en lista där varje krav utgör en rad. Listan kallas “product backlog”
- ✓ Processen förskriver inga specifika tekniker och metoder för utveckling.



SCRUM



Princip: Inga förändringar under en sprint



- ✓ En sprint pågår under så lång tid som det är praktiskt möjligt att stänga ute förändringar

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
- *Ceremonier/Aktiviteter*
 - Sprint Planning – planering i början av varje sprint
 - Sprint Review – ”redovisning” av resultatet från en sprint
 - Sprint Retrospective – ”sprint” post-mortem, förbättringsmöte
 - Daily Scrum Meeting – kort dagligt möte i utvecklingsteamet
- ✓ 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



Product Backlog

- ✓ En lista av önskat arbete på projektet
- ✓ Består oftast av en kombination av
 - “story”-baserat arbete, jmf. krav (“låt användaren söka bland produkter”)
 - Uppgiftsbaserat arbete (“förbättra systemloggning”)
- ✓ Listan ägs och prioriteras av Product Owner

- ✓ Kan även innehålla “Shores” – sysslor
 - ”Installera verktyg”
 - Backup på utvecklingsdatabasen.



Exempel på en Product Backlog

Prioritetsordning →

Korta beskrivningar →

Omfattning →

	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
		- 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



User stories

- ✓ “User stories” är enkla, kortfattade beskrivningar av en funktion.
- ✓ Historien ”berättas” utifrån samma perspektiv som den person som önskar den nya funktionen
- ✓ De följer vanligtvis en enkel mall .:

Som en <typ av användare>, vill jag <något mål>, så att <någon anledning>.

- ✓ User stories skrivs oftast på små lappar, exempelvis ”sticky notes”
- ✓ Detta underlättar i prioriterings och planeringsarbetet

- ✓ Motsvarar inte en traditionell kravspecifikation
- ✓ ”Ofullständiga” krav tills man börjar diskutera dem



Exempel

- ✓ **Som en förening vill vi kunna anmäla ett lag så att** det kan delta i tävlingen

- ✓ Vem?
- ✓ Vad?
- ✓ Varför?

- ✓ Hur kan vi lägga till detaljer?
 - Dela upp i flera mindre men mer detaljerade historier
 - Lägga till villkor. Vi är nöjda när.



Från mål till Sprint Backlog

- ✓ Utvecklingsteamet tar målsättningen med sprinten och bestämmer vilka uppgifter som krävs för att nå det.
 - Målet är ett urval av de user stories som finns i product backlog
 - Urvalet görs av utvecklingsgruppen
- ✓ Gruppen organiserar själva arbetet kring uppgifterna
- ✓ Backloggen byggs upp allt eftersom.



Sprint Backlog under en Sprint

- ✓ Förändringar
 - Teamet kan lägga till nya uppgifter som behövs
 - Ta bort onödiga uppgifter
 - **Viktigt!** Endast teamet kan förändra uppgifterna
- ✓ Förändra uppskattningarna vid behov allt eftersom

- ✓ Undvik för långa listor
- ✓ Bryt ned för stora uppgifter

- ✓ Varje uppgift har en egen status
- ✓ Uppdatera varje dag!



Sample Sprint Backlog

		Days Left in Sprint			
		15	13	10	8
Who	Description	7/22/2002	7/24/2002	7/26/2002	7/31/2002
Total Estimated Hours:		554	458	362	270
-	User's Guide	-	-	-	-
SM	Start on Study Variable chapter first draft	16	16	16	16
SM	Import chapter first draft	40	24	6	6
SM	Export chapter first draft	24	24	24	6
Misc. Small Bugs					
JM	Fix connection leak	40			
JM	Delete queries	8	8		
JM	Delete analysis	8	8		
TG	Fix tear-off messaging bug	8	8		
JM	View pedigree for kindred column in a result set	2	2	2	2
AM	Derived kindred validation	8			
Environment					
TG	Install CVS	16	16		
TBD	Move code into CVS	40	40	40	40
TBD	Move to JDK 1.4	8	8	8	8
Database					
KH	Killing Oracle sessions	8	8	8	8
KH	Finish 2.206 database patch	8	2		
KH	Make a 2.207 database patch	8	8	8	8
KH	Figure out why 461 indexes are created	4			



Roller



Product Owner

- ✓ Ansvarig för att definiera produktens funktionalitet – äger visionen
- ✓ Ansvarig för att besluta om releasedatum och innehåll
- ✓ Ansvarar för produktens “return of investment” (ROI)
- ✓ Prioriterar features beroende på dess relativa värde på marknaden
- ✓ Anpassar feature-listan och prioriteringarna efter behov i varje iteration.
- ✓ Accepterar resultat.

- ✓ Vem är det då?
 - Kan vara en användare av systemet
 - Någon från marknadsavdelningen
 - En ”produktansvarig” (product manager)



Product Owner – Egenskaper

- ✓ Tillgänglighet
 - En förutsättning för principerna i Scrum
 - En bra PO gör sitt yttersta för att produkten skall bli så bra som möjligt
- ✓ Kunniga inom produktens “affärsområde”
 - Ansvarar trots allt för vilka features vår produkt skall ha
 - Och prioriteringarna utifrån ”marknadsvärdet”
- ✓ Kommunikativ förmåga
 - Samarbete inom och utanför utvecklingsorganisationen
 - Olika ”språk” olika nivåer.



Scrum Master

- ✓ Motsvarar projektets ledning, **OBS inte projektledare!!**
- ✓ Hjälper teamet med Scrum *principerna* och att använda “*metoderna*” rätt
- ✓ *Tar bort hinder*
- ✓ Säkerställa att utvecklingsteamet *fungerar* och *producerar*
- ✓ *Möjliggör nära samverkan* mellan alla roller och funktioner
- ✓ Skydda utvecklingsteamet från externa ”*störningar*”



Scrum Master - Egenskaper

- ✓ Ansvarsfull
- ✓ Ödmjuk
- ✓ Samarbetsvillig
- ✓ Engagerad
- ✓ Kunnig
- ✓ Inflytelserik



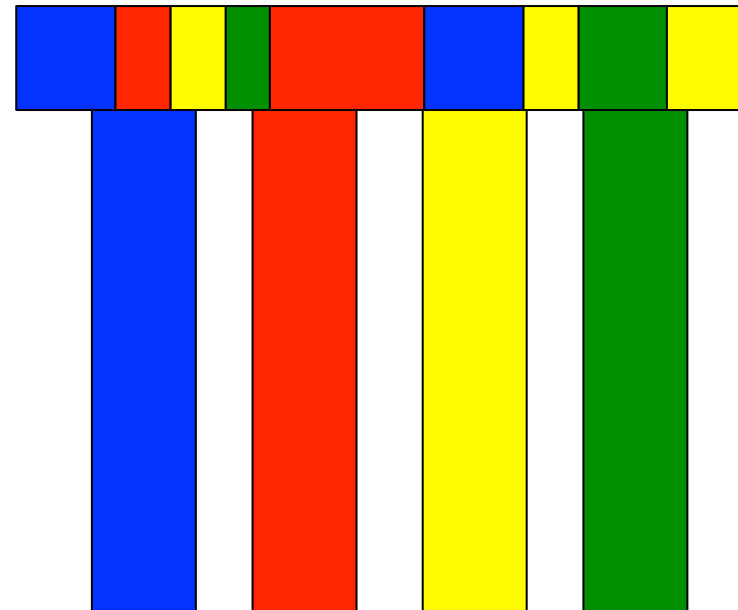
Utvecklingsteamet

- ✓ Vanligtvis 5-10 individer
- ✓ “Cross-functional”
 - Kvalitetssäkring,
 - Programmerare,
 - UI Designers, etc.
- ✓ Teamet organiserar sig själva
- ✓ Grupsammansättningen kan bara förändras mellan två sprintar
- ✓ Alla medlemmar är heltid i gruppen



“Cross-functional”

- ✓ Liten grupp individer som organiserar sig själva
 - ✓ Skall leverera körbar testad kod som kund utvärderar
 - ✓ Täcker in all ”metoder” och teknologin som behövs.
-
- ✓ Robusthet
 - ✓ Flexibilitet

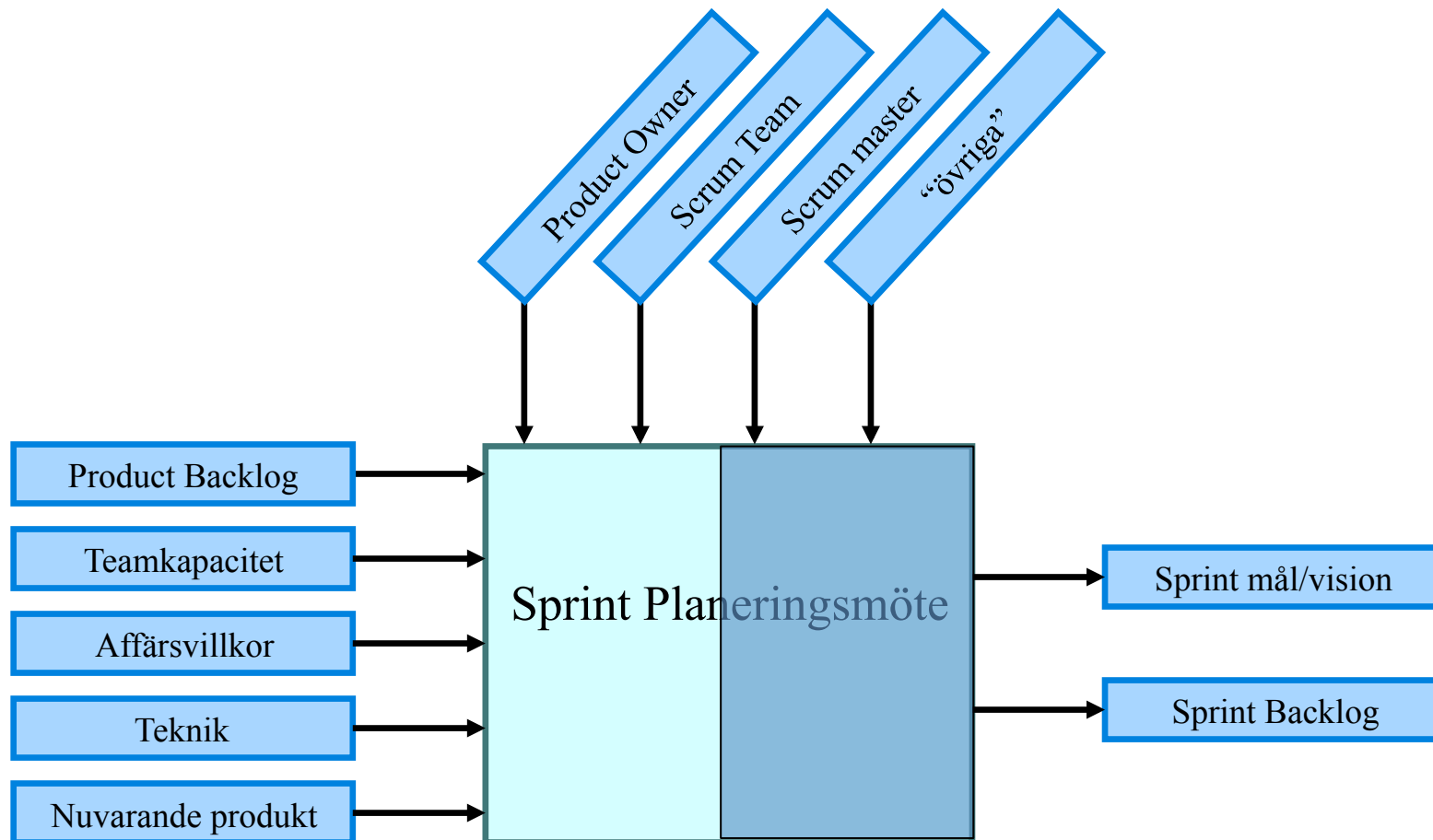


Ceremonier/Aktiviteter

- ✓ Sprint, planeringsmöte
- ✓ Sprint (vi går inte igenom denna i detalj)
- ✓ Daily Scrum
- ✓ Sprint, retrospektiv



Sprint planeringsmöte



De två stegen i ett planeringsmöte

- ✓ 1:a steget – med “kunden”
 - Skapa en “Product backlog”
 - Bestämna vilket mål man har för sprinten.
 - Deltagare i 1:a steget är
 - Product Owner,
 - Scrum Master,
 - Utvecklingsteamet
- ✓ 2:a steget – inom utvecklingsteamet
 - Skapa en ”sprint backlog”
 - Deltagare
 - Scrum Master
 - Utvecklingsteamet



Planeringsprinciper

- ✓ Utvecklingsteamet bestämmer själva i steg 1 och i steg 2!
- ✓ Kunden (PO) eller Scrum master kan inte påverka!

- ✓ Efter ett tag känner varje grupp hur mycket de kan “producera” i en sprint
- ✓ Detta kallas för *rytm*

- ✓ För att kunna *planera* och *mäta* en *grupps rytm* måste vi ha en “storlek” eller enhet. Scrum kan använda sig av
 - story points
 - “perfekta dagar”
- ✓ Viktigt med kända förutsättningar
 - Förändringar utifrån inte möjliga under sprinten
 - Detta är Scrum matterns ansvar



Planeringspoker (Planning poker)

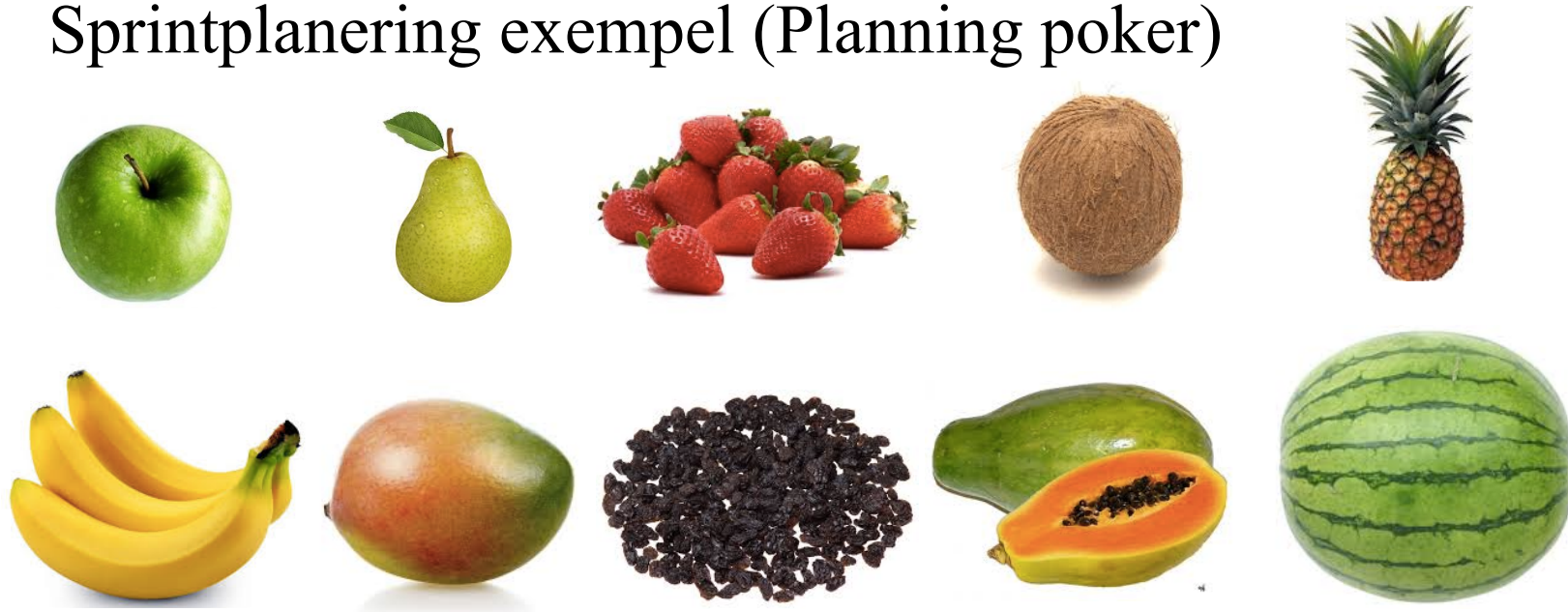
- ✓ Metod för att uppskatta omfattningen på en uppgift
- ✓ Bygger på konsensus inom en grupp

- ✓ För varje uppgift (exempelvis user story)
 - Varje medlem uppskattar omfattningen
 - Väljer ett kort och visar det.
 - Om några visar avvikande kort skall anledningen till detta diskuteras tills man når konsensus

0	½	1	2	3	5
8	13	20	40	100	?



Sprintplanering exempel (Planning poker)



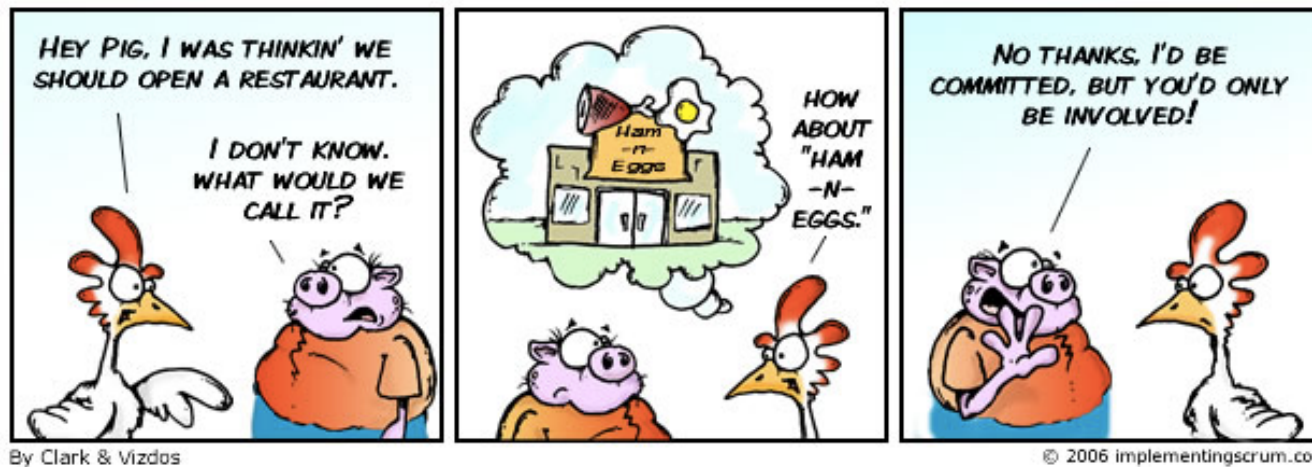
- ✓ En person skall göra en fruktsallad
- ✓ Det får ta maximalt 30 minuter
- ✓ ”Minimal Viable Product”
- ✓ Vilka frukter kommer med i fruktsalladen
 - Prioritet
 - Svårighet

Daily Scrum

- ✓ Parametrar
 - 15-minuter dagligen, “stand-up”
 - Skall inte lösa problem
- ✓ Tre frågor:
 - Vad gjorde du igår?
 - Vad skall du göra idag?
 - Vilka hinder finns i din väg?
- ✓ Kycklingar och grisar är inbjudna för att undvika onödiga möten
- ✓ Endast grisar får prata



Grisar och Kycklingar



- ✓ Grisar
 - Scrum master
 - PO
 - Utvecklingsteam
- ✓ Kycklingar
 - Intressenter
 - Ledning



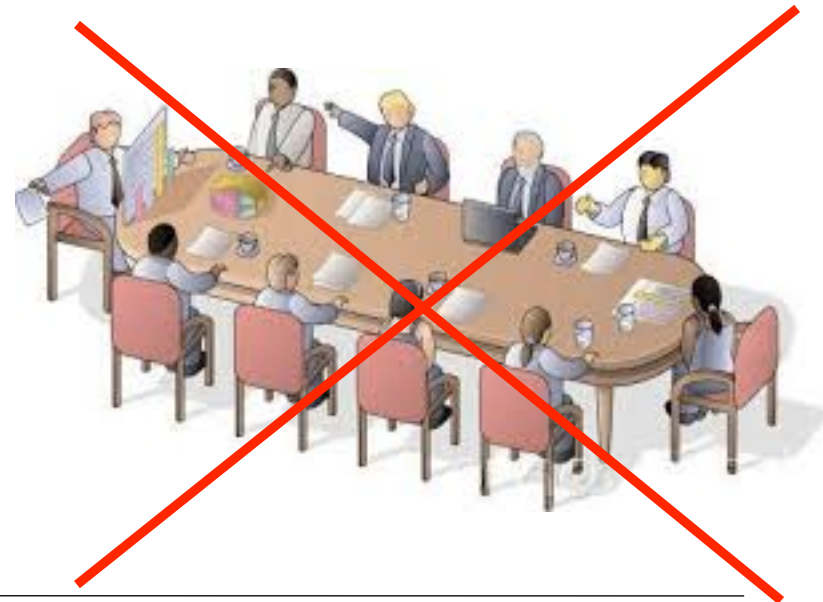
Daily Scrum

- ✓ Är INTE ett problemlösningsmöte
- ✓ Är INTE ett sätt att samla information om vem som ligger efter.
- ✓ Mötet är till för att teammedlemmar skall göra åttaganden inför varandra och för Scrum mastern
- ✓ Mötet ger Scrum mastern en bra överblick över projektets framsteg



Sprint Review Meeting

- ✓ Utvecklingsteamet presenterar vad de åstadkommit under sprinten.
- ✓ Vanligtvis en informell demo av features. Man talar om ”2hour prep”
- ✓ Återkoppling till PO.
- ✓ Deltagare
 - Kunder
 - Ledning
 - PO



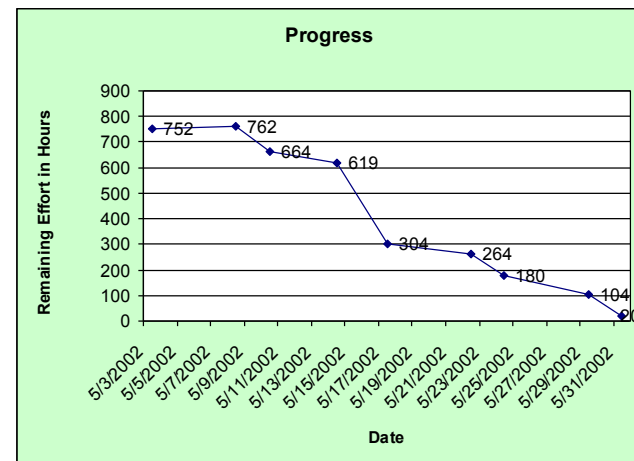
Övrig terminologi

✓ Definition of Done

- När är ett inkrement klart?
- Teamet måste fastlägga detta.
- Kopplat till kvalitet snarare än funktionalitet

✓ Sprint burndown chart

- Visar återstående timmar per dag till release
- Skall hamna på noll i slutet av en sprint



Metoder

- ✓ Scrum saknar beskrivningar av hur mjukvaran skall utvecklas
- ✓ eXtreme Programming beskriver totalt 12 agila “metoder”
 - 40 timmars vecka
 - User stories
 - Parprogrammering
 - Kontinuerlig integration
 - Test Driven Development
 - Kund på plats
 - Små leveranser

- ✓ Scrum innehåller en del av dessa.



Parprogrammering (Pair Programming)

- ✓ I ett par finns två roller som liknar de i en rallybil
 - ”*The driver*”, som fokuserar på att skriva syntaktiskt korrekt kod
 - ”*The navigator*”, funderar på hur den nuvarande produkten passar in i den övergripande designen.
- ✓ Inkrementell utveckling kräver disciplin och par-programmering säkerställer detta.
- ✓ Sättet att utveckla är blir mindre känsligt för externa störningar.
- ✓ Viktigast av allt. Det kräver att individer samarbetar och kommunicerar.



Kontinuerlig integration (Continuous Integration)

- ✓ Målsättningen för CI är att all kod skall kunna skickas till kund.
- ✓ Skall vi kunna leverera mitt I en sprint? Tanken är att det som produceras skall passa helheten även om all funktionalitet inte är på plats.
- ✓ CI gör att du endast integrerar mindre delar, vilket underlättar.
- ✓ Måste automatiseras
- ✓ Du måste testa och testerna ger dig grön ljus.



Kontinuerlig integration - Utmaningar

- ✓ Kräver en infrastruktur.
- ✓ Påverkar sättet att arbeta i hög grad
- ✓ Byggtiderna måste vara korta.



Slut på föreläsningarna

- ✓ Laboration 4
 - Planera 3 iterationer där ni skall
 - Förbättra/Förfina krav
 - Implementera
 - Testa
 - Reflektera

- ✓ Hemtentamen

