

Några ord om agila metoder

Mikael Kalms, EA DICE

Innehåll

Vattenfall

Agile development

eXtreme Programming (XP)

Scrum

Kanban

Lean

Vattenfall

Tämligen sekvensiellt...

Inhämtning av krav

Design

Implementation

Testning

Underhåll

Vattenfall - problem

Stora “iterationer” (v1.0, v2.0)
Kostnaden för ändringar kan bli stor

Originalversionen av vattenfall var inte tänkt att användas!

Agile development

Samlingsnamn för XP, Scrum och andra filosofier

XP

Funnits sedan sent 90-tal

Motreaktion mot rigida modeller (RUP, vattenfall)

Kent Beck
Ron Jeffries

XP

Minimera kostnaden för ändringar

Kontinuerlig dialog med kunden

XP

Design on-demand
Test Driven Development (TDD)
Parprogramming
Generalister, inte specialister

Ya Ain't Gonna Need It (YAGNI)

Planering iterationsvis (1-4 veckor)

XP - problem

Känns lite grand som en sekt

“fungerar inte XP för er, så gör ni inte rätt”

Ej välanpassat för fasta deadlines

Svårt göra TDD cross-team

Stora system behöver en del up-front design

Scrum

Lånar vissa grepp från XP

Mer fokus på planering, mindre om kodskrivandet

Funnits sedan tidigt 2000-tal

Scrum

Arbete i iterationer, “sprints”: ~2-6 veckor
Arbete i team: 3-10 personer

Roller inom ett team:

Produktägare
Scrum Master
utvecklare

Scrum: produktägarens uppgift

Skapa och underhåll en lista med features,
“backlog”

Sätta prioritet på dessa

Scrum: ScrumMasters uppgift

Sköta administrativt junk

Scrum: sprint

Ger utvecklarna fokus (ett fast mål) och frihet/svängrum under ett par veckor

1. Planera vad som ska göras
2. Implementera
3. Gör i ordning ett demo
4. Retrospective

Scrum: sprintplanering

Bestäm sprintens syfte

Ta reda på vilka utvecklare som finns tillgängliga

Bestäm vilka features som ska göras
+ forma Scrum-team

Scrum: bestämma vad som ska göras

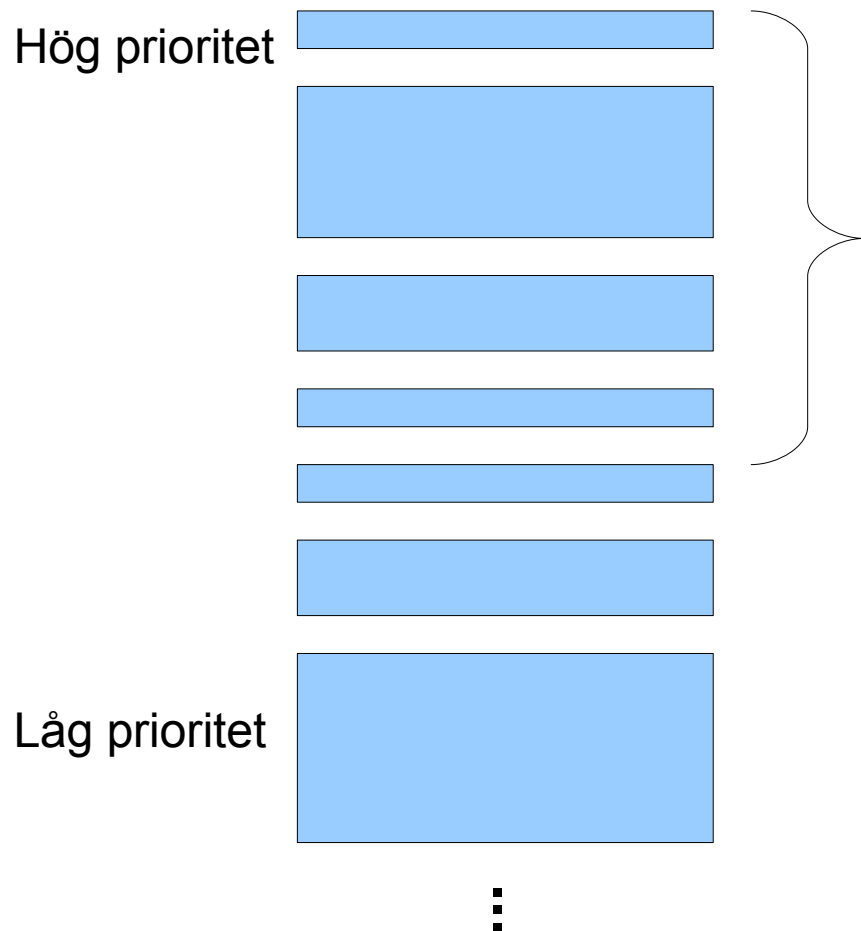
En feature har tre egenskaper

1. Scope
2. Prioritet
3. Estimerad tidsåtgång

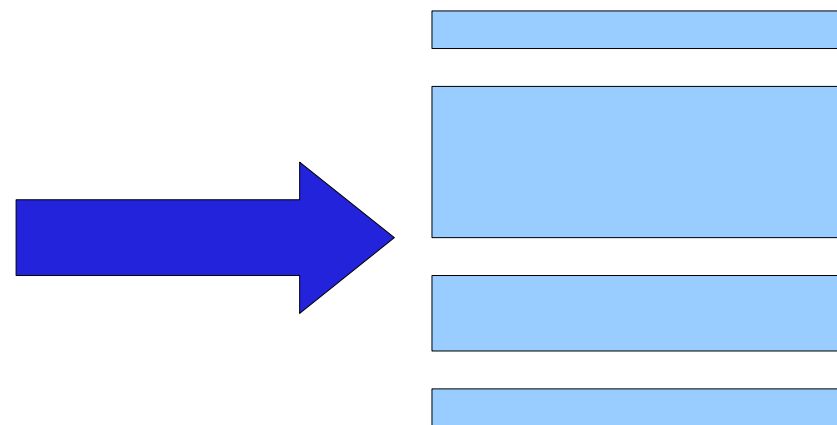
Produktägaren kontrollerar 1&2
Utvecklarna kontrollerar 3

Scrum: bestämma vad som ska göras

Produkt backlog



Sprint backlog



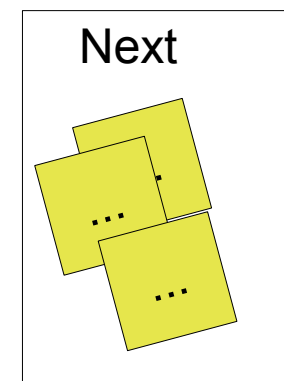
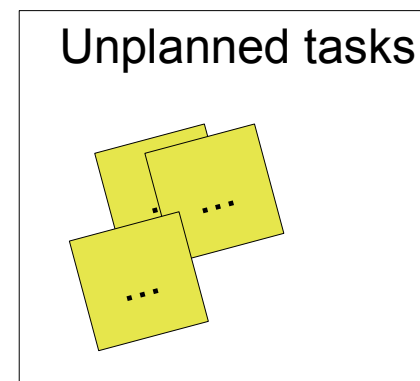
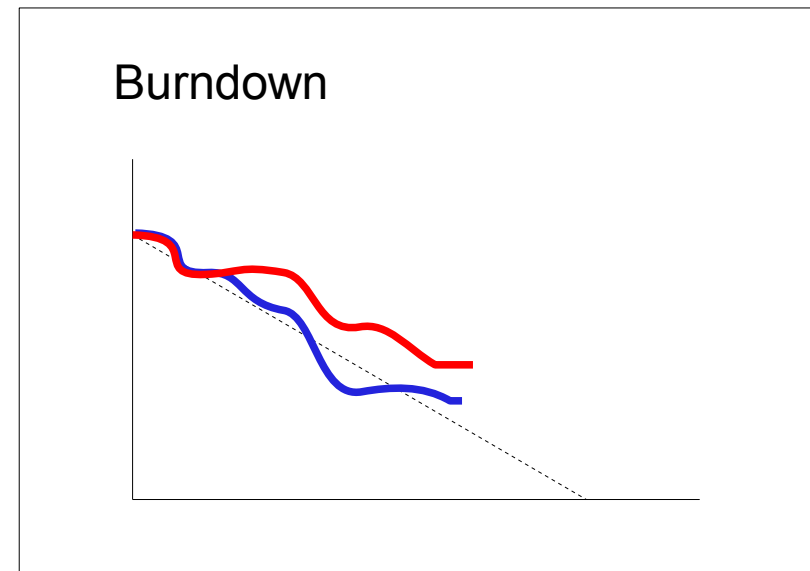
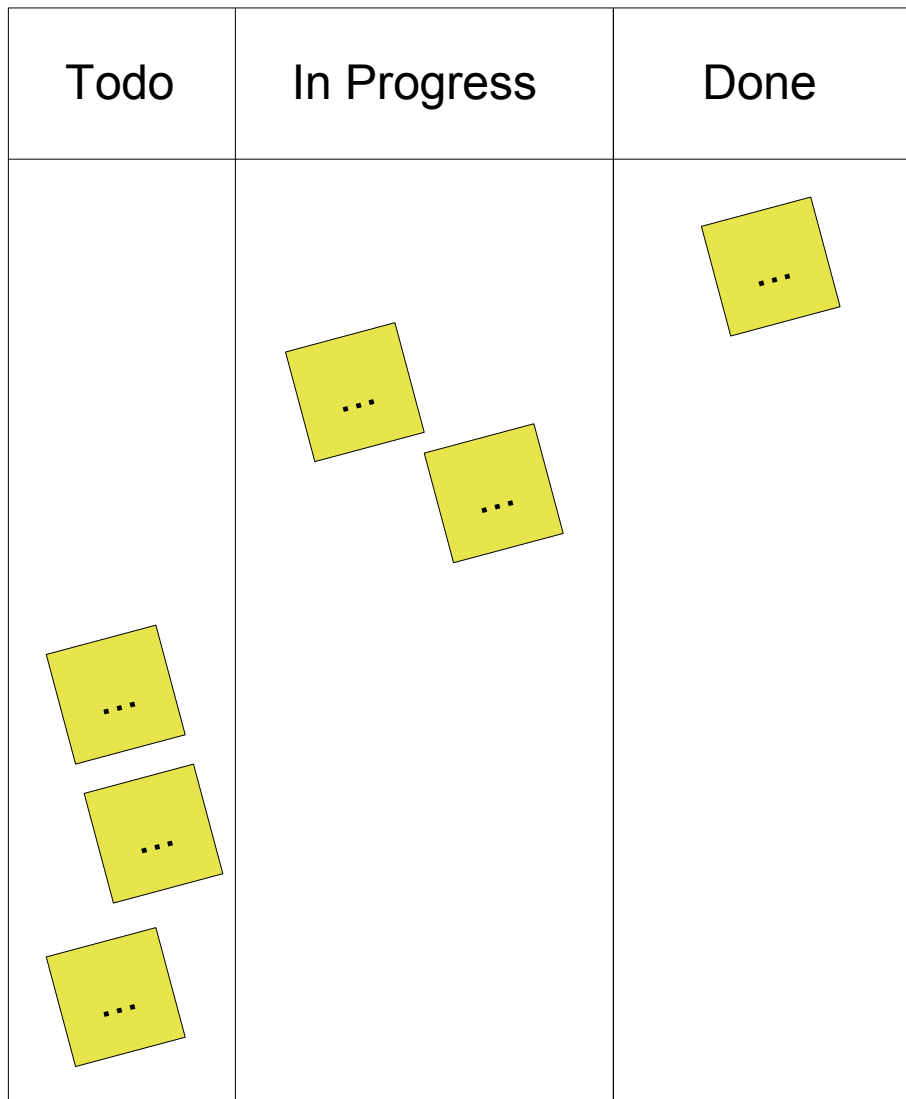
Scrum: implementation

Produktägaren har inget inflytande här

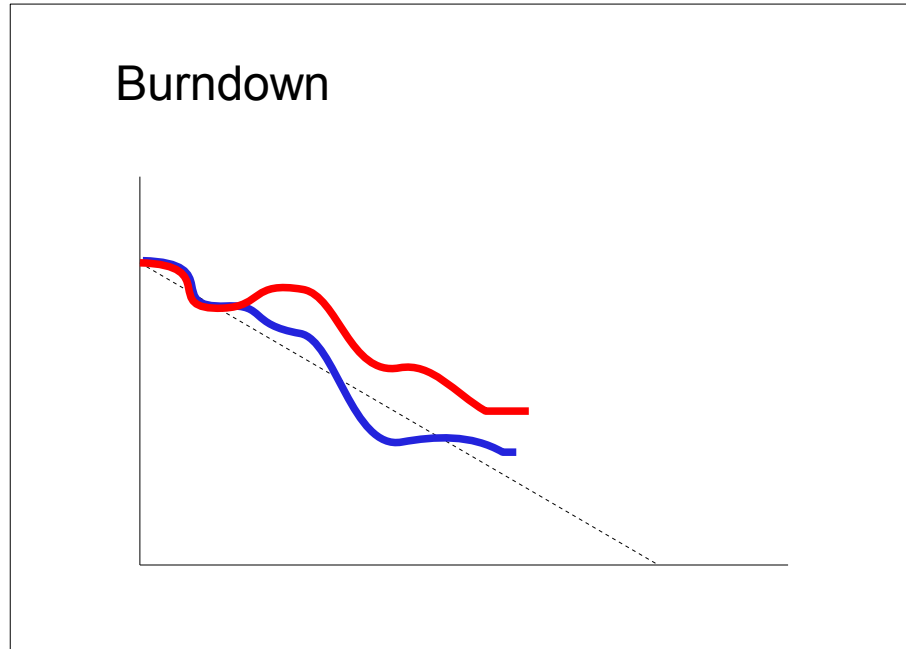
Beta av saker i prioritetsordning

Korta, dagliga möten

Scrum: en task board



Scrum: Burndown charts



Används för att mäta
hastighet

Streckad: ideal linje

Röd: färdiga tasks (enbart planerade)

Blå: färdiga tasks (planerade + oplanerade)

Scrum: mäta hastighet

Tre aspekter:

Hur snabbt betas planerade tasks av

Hur snabbt tillkommer oplanerade tasks

Hur snabbt betas oplanerade tasks av

mätningen är till för att upptäcka problem tidigt

Scrum: visa demo

Demonstrera alla nya features

Scrum: retrospective

Lufta tankarna.

Vad gick bra?

Vad gick dåligt?

Vad skulle kunna göras bättre?

Välj några få områden för förbättring till nästa sprint.

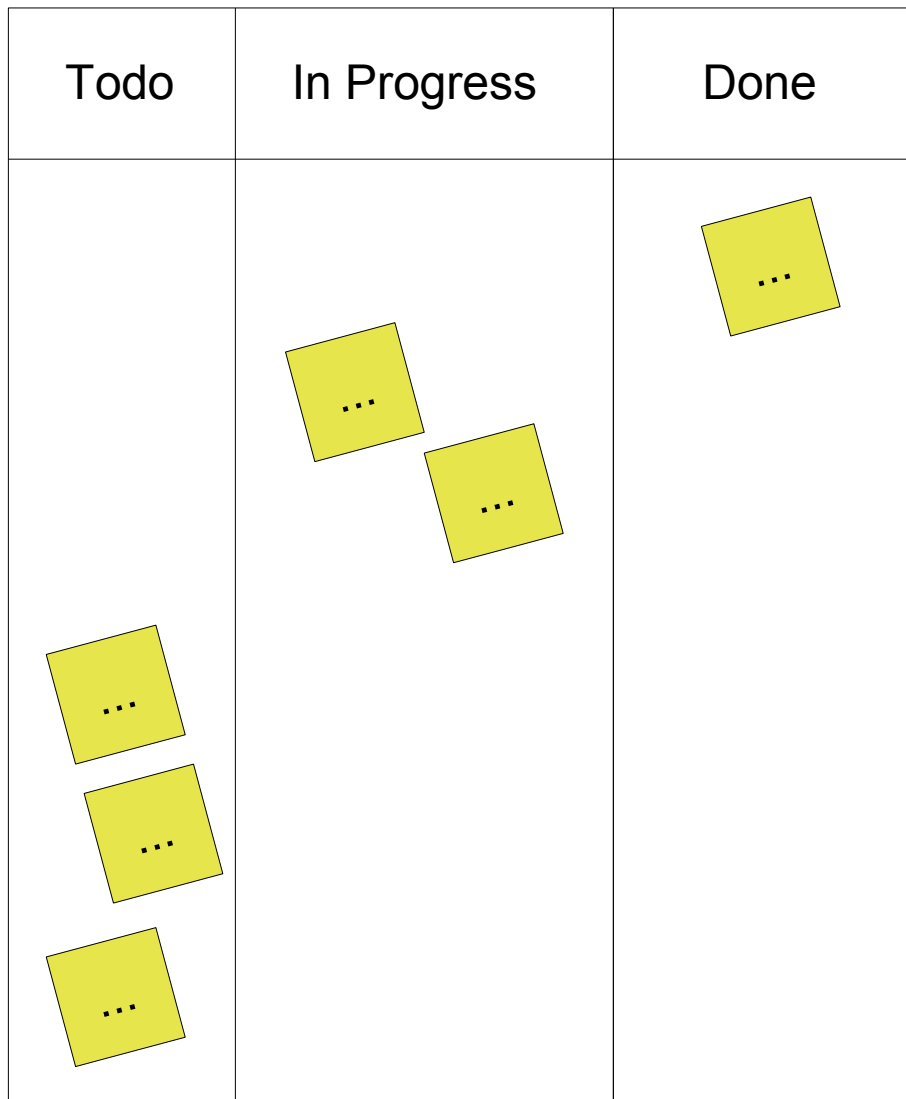
Scrum - problem

Produktägaren måste vara med på noterna

Lämpar sig bäst för nyutveckling, där slutgiltigt mål, total tid & scope inte är givna från början

Ingen long-term planering ingår i standardmodellen

Kanban – “pull systems”



Kommer från fordonsindustrin

Max antal items i “Todo” och “In Progress”

När det börjar bli tomt i “In Progress”, ta nya tasks från “Todo”

När det börjar bli tomt i “Todo”, ta nya från backlog

Mål: Eliminera onödigt arbete
Mål: Arbeta på få saker i taget

Kanban - problem

Svårt att utnyttja synergieffekter mellan tasks

Ingen tydlig “puls” i arbetet (som sprints/iterationer)

=> lämpar sig nog bättre för maintenance än för nyutveckling ... kanske ett alternativ när man buggfixar?

Lean development

Kommer från Toyota Production System

Huvudmål: eliminera “waste”

“waste” för oss: arbetstimmar som inte gör spelet
bättre

Lean: strategier

Försök få jämnt flöde

Försök korta ner cykeltider

och mkt mer :)

Further reading

Henrik Kniberg, “Scrum and XP from the
Trenches”

<http://www.infoq.com/minibooks/scrum-xp-from-the-trenches>

That's all folks

mikael@kalms.org