


Föreläsning 5: Processer och vidareutveckling

ETSA01 Ingenjörprocessen 1 - Metodik VT15 | Markus Borg



Detta har hänt....

- Pratat krav, plan, design, test
- På gång att frysa kravspecifikationen
- Övning 3+4: Test, partitioner, kodtäckning
- Börjat jobba med testplan och design
- Gjort förstudie kring implementationen?



Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörprocessen - Metodik

Agenda

Kursinformation

- L5b – Extern QA-granskning
- Om slutbedömning av projekt (acceptanstest)

Utvecklingsprocesser

- Linjära
- Evolutionära
- Inkrementella
- Lättrörliga (agila)

Programvara efter release

- Vidareutveckling
- Legacy-system



Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörprocessen - Metodik

Kursinformation

V 18:

- Mån kl 10 Föreläsning: Utvecklingsprocesser, vidareutveckling
- Mån kl 23:59: L4

V 19

- Mån kl 23:59: L5a (Testplan + Designdokument)

V 20:

- Mån kl 10 Föreläsning: Sammanfattning, utvärdering, om tentamen
- Mån kl 23:59: L5b (Extern QA-granskning)

V 22:

- Mån kl 23:59: Slutleverans projekt

V 23:

- Mån-Tis: Hementamen

Runt midsommar:

- Återkoppling och betyg



Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörprocessen - Metodik

L5b – Extern QA-granskning

QA = Quality Assurance = Kvalitetssäkring

- QA-ingenjörer är en egen yrkesroll inom software engineering
- Utformar arbetsprocesser
- Säkerställer att processer följs
- Utbildar medarbetare
- Kontrollerar att leverabler är av acceptabel kvalitet
- Kan vara verksamma som expertkonsulter

Er uppgift

- Kvalitetssäkra en annan grups testplan genom granskning



Lund University | Computer Science | Markus Borg | ETSAD1 Ingerjöprocessen - Metodik

L5b – Praktiskt utförande

Låt en annan grupp få insyn (L5a Måndag 4/5)

- Skapa en mapp på Google Drive "Kvalitetssäkring"
- Kopiera dit Kravspecifikation 1.X och Testplan 1.0
- Dela ut läs- och skrivrättigheter till en annan grupp

Kvalitetssäkra en annan grups testplan (L5b Måndag 11/5)

- Hitta er tilldelade grups mapp "Kvalitetssäkring"
- Granska deras Testplan 1.0
- Kravspecifikationen används bara som stöddokument
- Kopiera ett granskningsprotokoll till den andra gruppens mapp

Gruppfordelningen kommer på mail under veckan!



Lund University | Computer Science | Markus Borg | ETSAD1 Ingerjöprocessen - Metodik

Projektbeskrivning avsnitt 8: Acceptanstest i korthet

... uppfyller följande kriterier:

- Levererad programvara uppfyller projektets kravspecifikation
- Testas med projektets testfall + utforskande testning

Kravspecifikationens och testplanens kvalitet är god

- Kontrolleras med checklistor och projektets granskningar

Det är särskilt viktigt att alla kraven adresseras av minst ett testfall

Projektmodellen så som den presenterats följs i tillräcklig utsträckning

Kontrolleras genom:

- analys av levererade dokument: plan, krav, test, design, testrapport, granskningsprotokoll och installationsmanual
- verifiering av ändringshantering
- konsistens mellan slutversionerna av krav, design, test och applikation



Lund University | Computer Science | Markus Borg | ETSAD1 Ingerjöprocessen - Metodik

Vad händer i
maj/juni?

Projektets betyg
baseras på samtliga
dokument





1960-talet: "The Software Crisis"

Återkommande problem med programvaruprojekt

- Förseningar
- Många fel i levererade produkter
- Projekt som fick avbrytas
- Kostnader som skenade
- Produkter som bara blev större och större
- Etc.



Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörprocessen – Metodik

1960-talet: "The Software Crisis"

The major cause of the software crisis is that the machines have become several orders of magnitude more powerful! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem.

– Edsger Dijkstra, Turing Lecture, 1972



Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörprocessen – Metodik

Lösning: Styr upp röran!

NATO-konferens i Garmish-Partenkirchen 1968 myntade begreppet Software Engineering

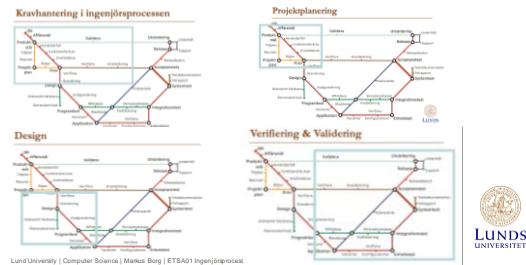
- Ta efter hur utveckling sker i andra ingenjördiscipliner
- Dags att införa "ordning och reda"
- God planering och uppföljning



Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörprocessen – Metodik

Omfattande processer följde...

Under 70- och 80-talet gällde uppstyrda processer och subprocesser



Många problemen kvarstod...

Det visade sig svårt att planera bort förändringar i kravbild

- Kunder ändrar sig
- Marknaden förändras
- Ny teknik införs hela tiden
- Hårdvaran byts ut
- Programvaruutveckling är innovation
 - Nya insikter följer hela tiden

Lund University | Computer Science | Markus Borg | ETSAD1 Ingenjörprocessen - Metodik

1990-talet: Motreaktion mot processerna

De kritiska rösterna:

- Programvara förändras hela tiden – går inte att planera bort denna osäkerhet
- Stryp inte utvecklingen med processer och dokument
- Utgå istället från ständiga förändringar
 - Flexibilitet är programvaras styrka

Lund University | Computer Science | Markus Borg | ETSAD1 Ingenjörprocessen - Metodik

Lättrörliga processer följde

Efterfrågas just nu...

Kompetenskrav:

- Systemutvecklare med mångårig erfarenhet av webbaserade system med Microsoft .NET som plattform, ASP.NET, C#
- Högskoleutbildning inom IT
- MS SQL Server databaser, databasmodellering och SQL
- HTML5 och CSS3
- **Agil** eller liknande agilt metodik

Kompetens:

- Du är civil- eller högskoleingenjör med några års erfarenhet av mjukvaruutveckling i tekniska tillämpningar
- Du har erfarenhet av **Scrum, Extreme programming, eller** liknande
- Du är väl förtrogen med **Agil** eller liknande metodik
- Det är extra meriterande om du har erfarenhet inom energi, verkstads- eller fordonsindustrin
- Som person är du utåtriktad, social och ansvarsfull. Du gillar att arbeta i team och har ett stort tekniskt intresse.

Meriterande

LIVSMEDLEVERNET

Jobbat med upphandling av IT-tjänster inom offentlig verksamhet.
ITIL-ramverk eller motsvarande.

Du har förmåga att arbeta mål- och resultatorienterat samt operativt och lösningsförmåga.

Erfarenhet av Kanban är meriterande.

Lund University | Computer Science | Markus Borg | ETSAD1 Ingenjörprocessen - Metodik

Framtiden: Utmaningar återstår

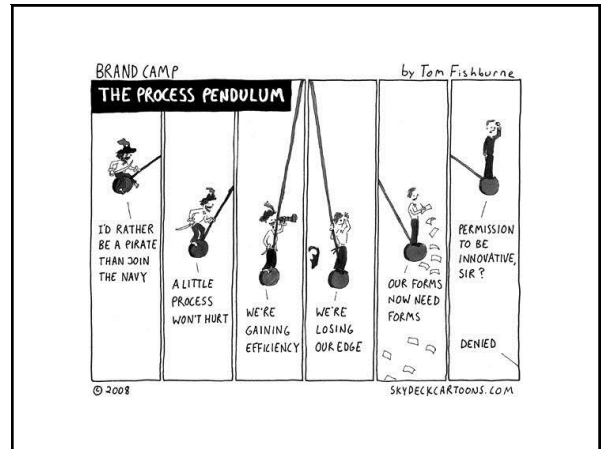
Öppna forskningsfrågor

- Hur arbetar 100/500/1000 ingenjörer lätttröligt ihop?
- Hur arbetar man lätttröligt i globala projekt?
- Kan man utveckla säkra programvara med lätttröliga processer?
- Hur underhåller man ett system utan dokumentation?

Kritiska röster pekar på misslyckade lätttröliga projekt



Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörsprocessen - Metodik



Processer
- Centrala begrepp

ETSA01 Ingenjörsprocessen 1 - Metodik VT15 | Markus Borg

Utvecklingsprocesser - begrepp

Process:

- Det arbete som görs för att utveckla programvara

Processmodell:

- En beskrivning av processen

Processförbättring:

- Arbetet med att förbättra processen



Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörsprocessen - Metodik

Aktiviteter som ingår i alla processmodeller

Specifikationsaktiviteter
 Utvecklingsaktiviteter
 Verifieringsaktiviteter
 Vidareutvecklingsaktiviteter

... men

- kan komma i olika ordning
- ha olika fokus



Lund University | Computer Science | Markus Borg | ETSAD1 Ingeringsprocessen - Metodik

Processer är antingen

Aktivitetorienterade

- Förstudie
- Kravinsamling
- Design
- Implementation
- Testning

Resultatorienterade

- Rapport från förstudie
- Kravspecifikation
- Design
- Källkod
- Testrapport



Lund University | Computer Science | Markus Borg | ETSAD1 Ingeringsprocessen - Metodik

Ingår i en processmodell

(Sub)processer – en serie steg med gemensamt mål

Aktiviteter

Leverabler

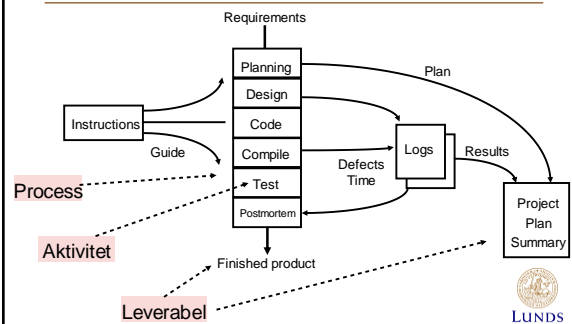
Villkor – startvillkor, slutvillkor

Roller

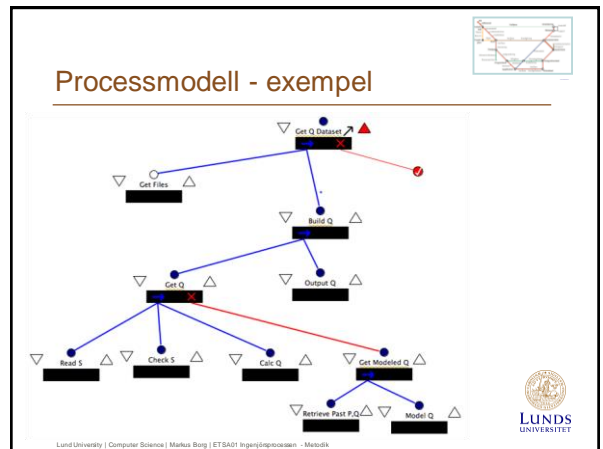
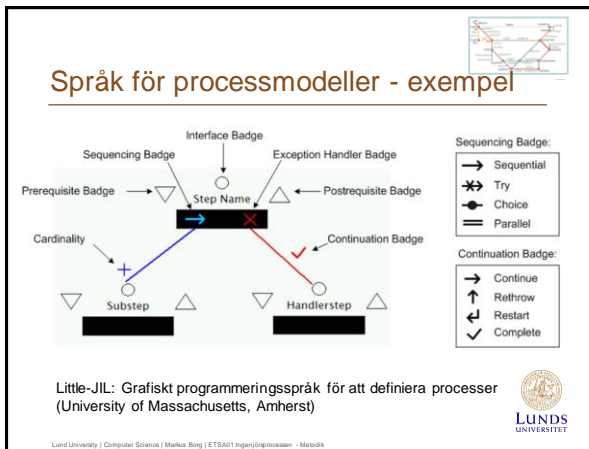
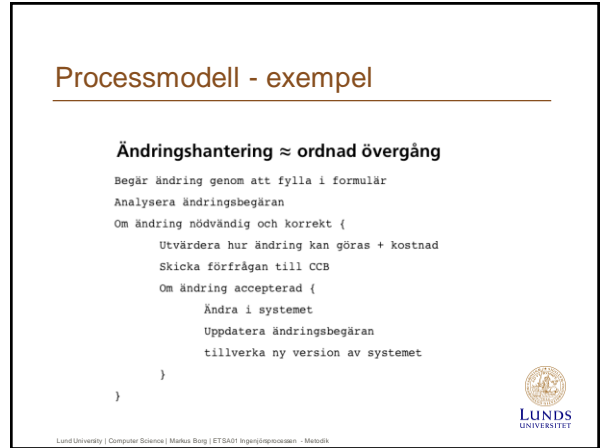
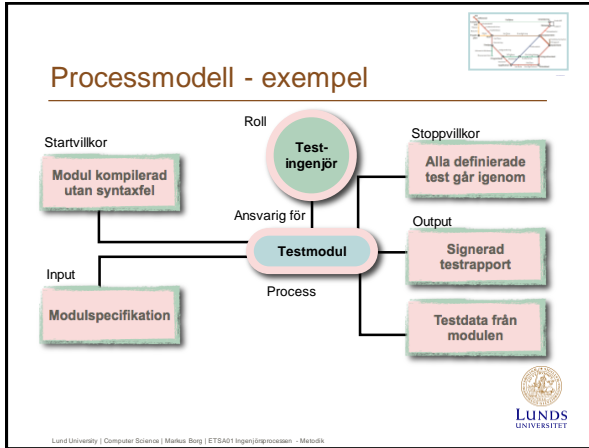


Lund University | Computer Science | Markus Borg | ETSAD1 Ingeringsprocessen - Metodik

Processmodell - exempel



Lund University | Computer Science | Markus Borg | ETSAD1 Ingeringsprocessen - Metodik



Varför engagera sig i processer?



Kvalitet

- Stark koppling mellan processkvalitet – produktkvalitet
 - Utvärdera processen kan vara enklare än att utvärdera produkten – och mer bestående
- En obefintlig process
 - kan leda till en utmärkt produkt
 - kan bli ett fullständigt misslyckande
- En bra process
 - synliggör problem och erbjuder åtgärder

Spara pengar

- En tydlig process minskar förvirring
- Processer kan återanvändas i andra projekt



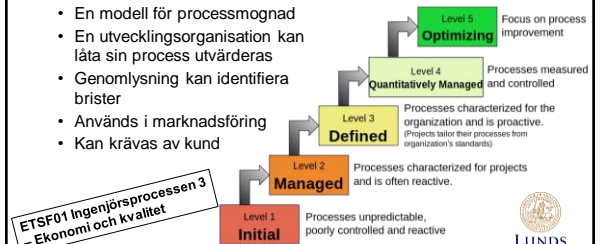
Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörprocessen - Metodik

CMMI – En modell för processmognad

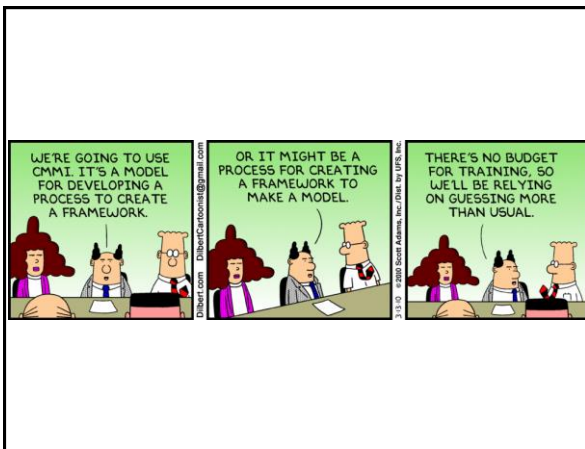


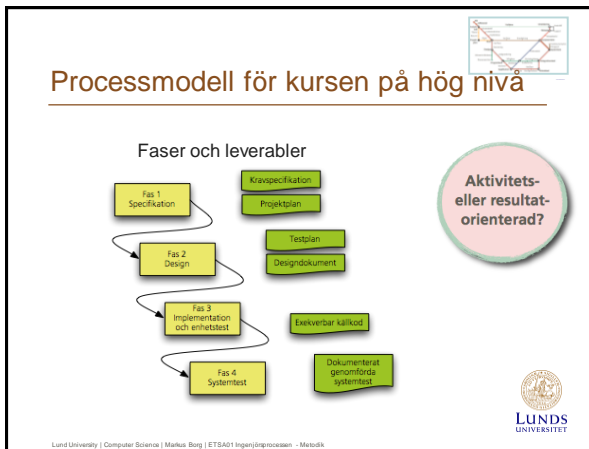
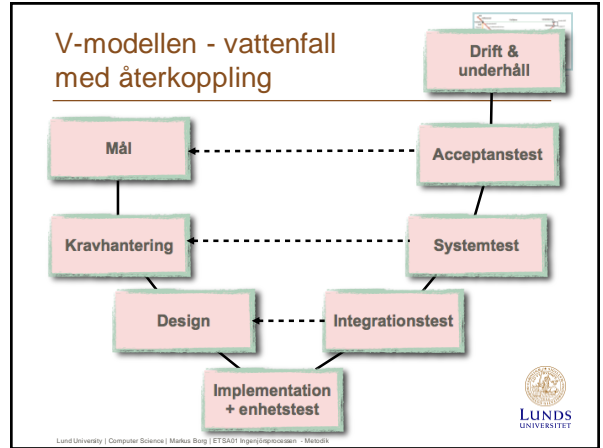
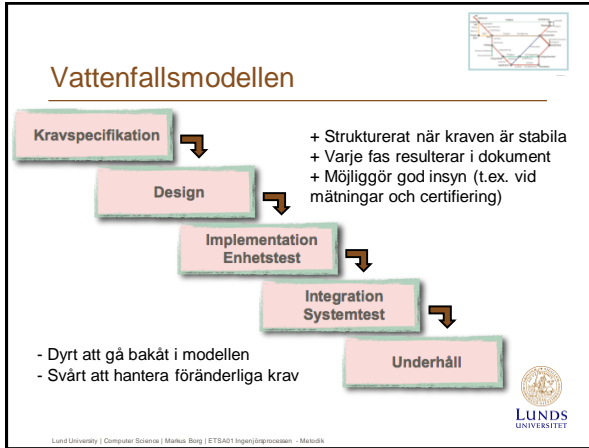
Capability Maturity Model Integration - CMMI

- En modell för processmognad
- En utvecklingsorganisation kan låta sin process utvärderas
- Genomlysning kan identifiera brister
- Används i marknadsföring
- Kan krävas av kund



Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörprocessen - Metodik





Linjära modeller: Problem

Ofta kan kraven inte frysas

Big bang-integration sällan framgångsrik

Ökar risk för "nice-to-have" bland kraven för säkerhets skull

Kostnaden för att underhålla dokument är hög - kundnytta?

LUND UNIVERSITY

Evolutionär utveckling

Utvecklingsprocessen

Mål

Rapport Utvärdering Behov Uppdaterad Utvärderad

Test Design Beslut Krav

Versioner för feedback

Kund / användare

Slutgiltig version

LUND UNIVERSITET

Två typer av evolutionär utveckling

Kasta-bort prototyping

- Ta fram prototyp, dra lärdomar, kasta bort
- Därefter utveckla "riktigt" system
- Används vid stora risker med tekniken
- Börja med svåra tekniska krav

Utmaningar
Svag insyn
Kan ge dålig struktur
Höga krav på utvecklarna

Evolutionär prototyping

- Arbeta hela tiden med "riktigt" system
- Används vid stora risker med kraven
- Börja med kundens viktigaste eller osäkraste krav

LUND UNIVERSITET

Spiralmodellen

Cumulative cost

Progress

1. Determine objectives

2. Identify and resolve risks

3. Development and Test

4. Plan the next iteration

Review

Requirements plan

Risk analysis

Risk analysis

Operational Prototype

Draft

Detailed design

Code

Integration

Test

Implementation

Release

Verification & Validation

Test plan

Den tydliga skillnaden är explicit riskhantering i varje varv!

LUND UNIVERSITET

Inkrementell utveckling

Fokus på systeminkrement

Möjliggör parallell utveckling

Ger tidiga leverabler

Möjlighet att identifiera krav under utvecklingen

Tidiga inkrement kan testas många gånger

Krav


Impl. Test Release

Impl. Test Release

Impl. Test Release


LUND UNIVERSITET

Timeboxing




Kravteam Kravställning
 Utveckling Implementation
 Testare Test

- Slöseri med resurser?



Lund University | Computer Science | Markus Borg | ETSAD1 Ingerjöprocessorn - Metodik


Timeboxing



- Specificera längd på en timebox, t.ex. 2 veckor
- Bryt ned projektet i lika stora arbetspaket


2 veckor

Krav	Impl.	Test
Krav 1	Impl. 1	Test 1
Krav 2	Impl. 2	Test 2
Krav 3	Impl. 3	Test 3




Lund University | Computer Science | Markus Borg | ETSAD1 Ingerjöprocessorn - Metodik

Timeboxing




Kravteam	Krav 1	Krav 2	Krav 3
Utveckling	Impl. 1 Impl. 2 Impl. 3		
Testare	Test 1 Test 2 Test 3		

Release 1 Release 2 Release 3




Lund University | Computer Science | Markus Borg | ETSAD1 Ingerjöprocessorn - Metodik

Jämför med pipeline-arkitektur i CPU



		Clock Cycle								
		0	1	2	3	4	5	6	7	8
PIPELINE	Waiting Instructions									
	Stage 1: Fetch									
	Stage 2: Decode									
	Stage 3: Execute									
	Stage 4: Write-back									
Completed Instructions										



Lund University | Computer Science | Markus Borg | ETSAD1 Ingerjöprocessorn - Metodik

Manifesto for Agile Software Development

(2001)

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Lättrörliga utveckling (agile)



Kärnan i agila processer

- Individer och interaktion > processer och verktyg
- Fungerande programvara > fullständig dokumentation
- Samverkan med kund > kontraktsförhandlingar
- Anpassning till förändringar > följande av plan



Lund University | Computer Science | Markus Borg | ETS&O Ingenjörsprocessen - Metodik

Vad innebär det att vara lättrörlig?



- Direkt kommunikation (face-to-face)
 - Mindre dokumentation
- Team styr själva sitt arbete (self-organizing)
 - Mindre toppstyrning
- Team med blandade kompetenser (cross-functional)
 - Krav, kod, test i samma team
- Adaptiv planering
 - Mindre förutspå framtid, mer anpassning till verklighet
- Evolutionär utveckling
- Kontinuerlig förbättring
- Tidiga releaser med snabb feedback från kund



Lund University | Computer Science | Markus Borg | ETS&O Ingenjörsprocessen - Metodik

kursur agil utveckling

Webben Bilder videor Kartor Nyheter Fler Sökverktyg

Ungefär 26 900 resultat (0.47 sekunder)

Kurs för Scrum Masters - citerus.se
www.citerus.se/scm *
Går som över 1000 leverantör sedan 2006 - boka din plats nu

Kurser i Agila metoder - Lean, Scrum, Kanban, test, UML
www.informatik.se/kurser/agila-metoder *
Boka din kurs enkelt online
Microsoft Gold Partner - Infrastruktur - Systemutveckling - Kontakt oss

Utbildning inom Scrum - dfkompetens.se
www.dfkompetens.se/ledning/agile *
Utbildningar inom Agile & Scrum Skapa affärsvärde med suveränt
Certifierande program - Kurser - Konferenser - Informationsmöten
Kontakt oss - Föreläsningsutbildningar - Certifierande program

Agile kurs & utbildning - Informatör
informatör.se/utbildning/agile-metoder *
Agile kurs Agila metoder har länge varit ett flitigt använt begrepp i utvecklarbranschen.
Möter som t.ex. Scrum och Kanban har skapat ett helt nytt sätt att utveckla ...

Ledarskap för agil utveckling - Management 3.0 | Sothouse
sothouseeducation.com - Kurser *
Om kursen Ledarskap för agil utveckling. Ledarskap för agil utveckling i
Management 3.0 cover. Har du som chef eller ledare någonstans undrat hur man
leder agil ...

Förbättrad agil utveckling för Scrum-team | Kurs 1812
www.learnagile.se/courses/_forbattrad-agil-utveckling-for-scrum-team/
På den här kursen lär du dig att lösa organisatoriska problem som hindrar
utvecklingen av dina agila team, skapa högrepresterande team för bättre ...
24 jan - 26 jun Stockholm
29 jul - 31 okt London

Agil Utveckling
www.attentec.se/Agil-utveckling -
Hastigt programutveckling.
Specialister på agila metoder!

Kurs Agil Kravanalytiker
www.inceptive.se/agilkravanalytiker *
Så är det hela rätt i det agila
- Då har vi kursen för dig!

Kurser för yrkesverksamma
www.utbildning.se/ *
Sök efter kompetensutveckling
- För dig i arbetslivet


Scrum Product Owner
www.rtf.se/ *
Så står du en utveckling med hjälp
av Scrum

Agil projektledning
www.projectcompassion.se/ *
Hantera sanning och spänning!
Orätta i en projekt!

Vissa din annons här >


<ul style="list-style-type: none"> Agil HR Workout Nyhet 	1 dag	9 800 kr	Läs mer och boka >
<ul style="list-style-type: none"> Agil Kravhantering 	1 dag	9 800 kr	Läs mer och boka >
<ul style="list-style-type: none"> Lean Agile Change Management Nyhet 	2 dag	18 600 kr	Läs mer och boka >
<ul style="list-style-type: none"> ISTQB Agile Tester 	2 dagar	13 500 kr	Läs mer och boka >
<ul style="list-style-type: none"> Agil Modellering med UML 	3 dagar	24 500 kr	Läs mer och boka >
<ul style="list-style-type: none"> Användbarhet & Agil användarcentrerad design Nyhet 	3 dagar	24 500 kr	Läs mer och boka >
<ul style="list-style-type: none"> Pm³ och Agila metoder Nyhet 	2 dagar	14 500 kr	Läs mer och boka >

Extreme Programming (XP)




- Första agila processmodellen som fick stor spridning
- Tekniskt fokus snarare än övergripande process
- Består av 12 praktiker (practices) att följa

Planning Game	Pair Programming
Small Releases	Collective Ownership
Metaphor	Continuous Integration
Simple Design	40-hour week
Testing	On-site Customer
Refactoring	Coding Standard



Lund University | Computer Science | Markus Borg | ETS260 Ingerjöprocessen - Metodik

Karaktäristiska praktiker i XP



Planning game

- All utveckling utgår ifrån en "user story" som kunden vill uppnå

Collective ownership

- Ingen äger någon specifik kod, alla arbetar med allt

Refactoring

- Koden uppdateras kontinuerligt för att förbättras

Testing

- Först skriver man testfallet, sedan koden som uppfyller det

Parprogrammering


- All kod skrivs i par av programmerare. Granskas i realtid.

Continuous integration

- Kod stoppas kontinuerligt in i huvudsystemet. Ingen big bang.


On site customer

- Kunden ska vara tillgänglig under hela utvecklingen för feedback



Lund University | Computer Science | Markus Borg | ETS260 Ingerjöprocessen - Metodik

XP: Brister och kritik




- En utvecklingsprocess ska inte detaljstyra hur utvecklarna programmerar
 - Parprogrammering är en vattendelare!
- Ofta omöjligt att ha en kund på plats
- XP är inte lika populärt som för 10 år sedan

Mer om detta i Programvaruutveckling i grupp - projekt ETS260

Mitt tips: XP innehåller bra idéer

- Se de 12 praktiker som ett smörgåsbord
 - Ta det ni gillar!



Lund University | Computer Science | Markus Borg | ETS260 Ingerjöprocessen - Metodik

Scrum



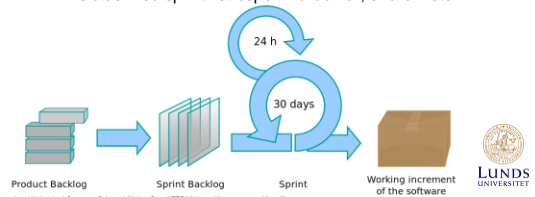
(Wikimedia commons: PierreSelim)



Lund University | Computer Science | Markus Borg | ETSAD1 Ingeröjningsklassen - Metodik

Scrum - huvudprinciper

- Systemkraven hanteras i en **product backlog**
- Utvecklingsarbetet bryts ned i **sprintar** (1-4 veckor)
 - Varje sprint innebär ett inkrement av systemet
 - Sprintens krav hanteras i **sprint backlog**
 - Inleds med sprint-planering: vad, vem, hur?
 - Avslutas med sprint-retrospekt: lärdomar, erfarenheter?



Daily scrum: Stående morgonmöte på 15 min

1. Vad gjorde du igår?
2. Vad ska du göra idag?
3. Ser du några hinder i ditt arbete?



(Wikimedia commons: flierce)

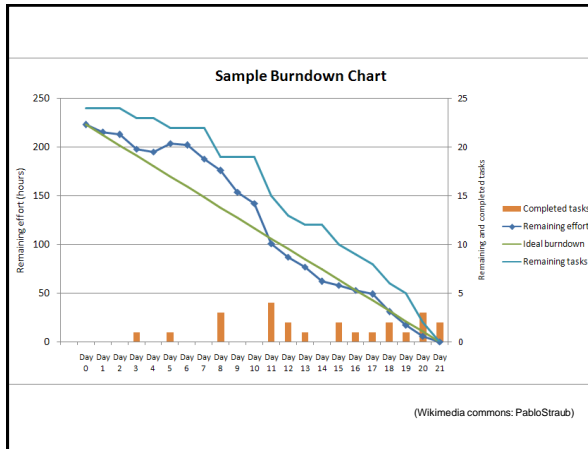
Scrum task board

Aktivitetsöversikt:

- Att göra
- Pågående implementation
- Pågående testning
- Färdigt



(Wikimedia commons: flierce)



Scrum: Brister och kommentarer

- Utveckling är svårt trots att Scrum används

"75% of organizations using Scrum will not succeed in getting the benefits that they hope for from it"

- Ken Schwaber (en av skaparna)

- Svårt att skapa Scrum-teams
 - Nyckelpersoner kan behöva hoppa mellan team
 - Passar dåligt för distribuerad utveckling



Lean software development



Filosofin "lean" produktutveckling utvecklades av Toyota

- Fokusera på att eliminera slöseri
- Gör det uppenbart att allt arbete leder till kundnytta genom att ta bort allt annat

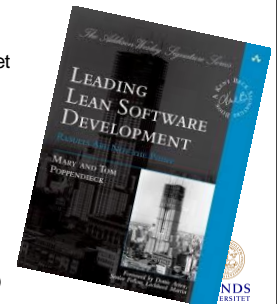


Lean software development

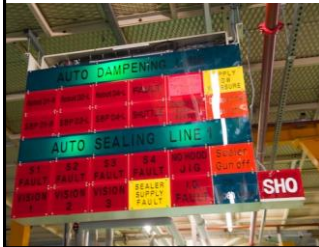
Slöseri (waste):

- Onödig kod och funktionalitet
- Otydliga krav
- Byråkrati
- Långsam kommunikation
- Fördröjningar i arbetet

(Mary Poppendieck and Tom Poppendieck, 2002)



Kanban



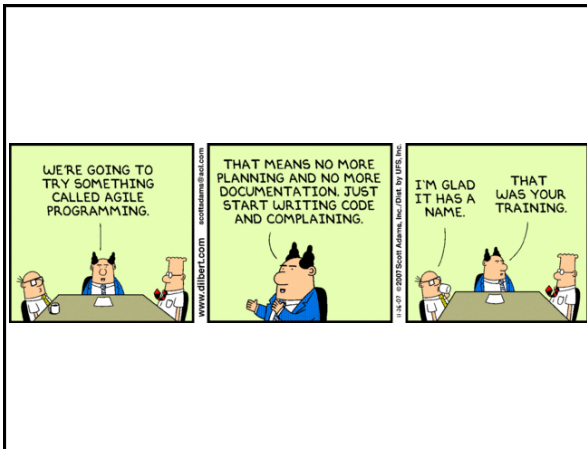
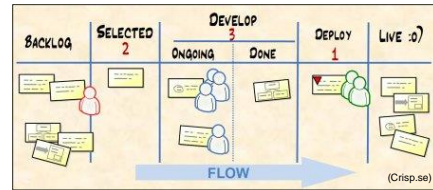
- Betyder ungefär "visuellt tecken" på japanska
- Också från Toyota
- Arbetare på löpande bandet signalerade tydligt resultat och behov
- Schemaläggning ger stöd åt lean och "just-in-time" produktion



Kanban för programvaruutveckling

Kärnan i Kanban

- Visualisera arbetsflödet – använd Kanban-tavla
- Begränsa "work in progress" – fokusera på genomströmning (flow)
- Mät ledtid för aktiviteter – sträva efter kontinuerlig förbättring



Finns ingen fulländad process!

No Silver Bullet

Lund University | Computer Science | Markus Borg | ETBAGI Ingeringsprocessen - Metodik



Programvara efter release

ETSA01 Ingenjörprocessen 1 - Metodik VT15 | Markus Borg




Nyutveckling vs. vidareutveckling

- Det mesta utvecklingsarbetet är vidareutveckling av befintliga system
- Software evolution innebär att anpassa programvaran efter release
- Underhåll av programvara handlar inte om slitage, istället:
 - Hantera nya krav
 - Porta till nya plattformar
 - Integrera med andra system
 - Rätta buggar
 - Optimera
 - Etc.



Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörprocessen - Metodik

Legacy system

"A legacy system is an old [method](#), [technology](#), [computer system](#), or [application program](#) that continues to be used, typically because it still functions for the users' needs, even though newer technology or more efficient methods of performing a task are now available."




Ofta affärskritiska

- Annars hade de inte funnits kvar
- Kan vara gamla (ibland > 30 år)
- Ingen har längre förståelse för system/arkitektur/teknik



Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörprocessen - Metodik




CARL QUIT. HE'S THE ONLY ONE WHO KNOWS HOW TO PROGRAM THE LEGACY SYSTEM.

IT CAN'T BE THAT HARD. GO FIGURE IT OUT.

FRACK

www.dilbert.com

© 1995 Scott Adams. All Rights Reserved. By NEA, Inc.



Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörprocessen - Metodik

Lehmans Lagar om software evolution

Totalt åtta "naturlagar" om vidareutveckling av programvara

De två viktigaste:

Kontinuerliga krav på förändring

- Så länge ett program används så kommer det att behöva uppdateras för att förbli uppskattat

Ökande komplexitet

- När man kontinuerligt ändrar i ett program blir det mer komplext och svårare att förstå – om man inte aktivt motverkar det



Lund University | Computer Science | Markus Borg | ETSAD1 Ingerjöprocossen | Metodik

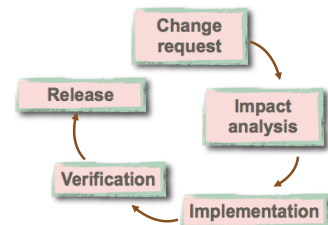
Underhåll efter release

Kräver

- Ändringshantering
- Konfigurationshantering
- Påverkansanalys
- Spårbarhet
- Regressionstest

Man skiljer mellan:

- Felrättande (Corrective)
- Anpassande (Adaptive)
- Förbättrande (Perfective)
- Förebyggande (Preventive)



Lund University | Computer Science | Markus Borg | ETSAD1 Ingerjöprocossen | Metodik

Vilka är viktigast?
Hur påverkar rätningen systemet?
Vem ska rätta den?

Sammanfattning – föreläsning 5

Utvecklingsprocessen är det som görs för att utveckla programvara

Processmodellen är en beskrivning av processen

Några exempel på processmodeller:

- vattenfall
- prototyputveckling
- spiralmodellen
- timeboxing
- lättrollig utveckling (agile)

Agila "buzz words"

- XP, scrum, lean, kanban

Legacy system är gamla system som levt kvar länge, i många fall längre än vad man förväntade sig från börja

Underhåll av programvara är antingen:

- felrättande
- anpassande
- förbättrande
- förebyggande

Underhåll underlättas av tydlig ändringshanteringsprocess, spårbarhet, konfigurationshantering och regressionstestning

Lund University | Computer Science | Markus Borg | ETSAD1 Ingerjöprocossen | Metodik