LUNDS UNIVERSITET

Föreläsning 6: Vidareutveckling, konfigurationer, produktledning

Programvaruutveckling - Metodik 2018 | Markus Borg

Agenda F6

1. Kursinformation
   - Status i projekten
   - Veckans leverabler
2. Ordinarie kursinnehåll
   - Vidareutveckling
   - Konfigurationer och versioner
   - Produktledning
3. Snabbgenomgång med feedback

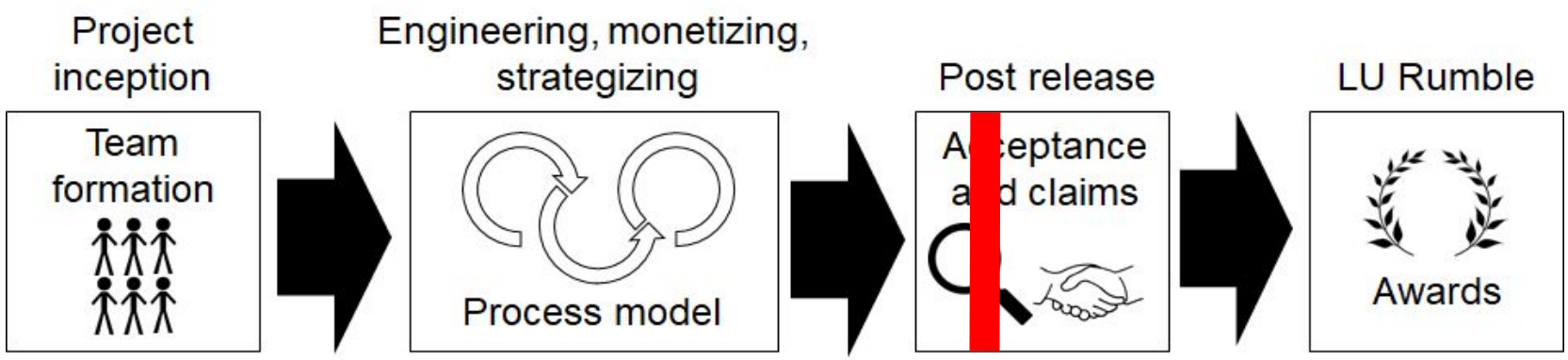
LUNDS UNIVERSITET

LUNDS UNIVERSITET

Robotprojekten

Programvaruutveckling - Metodik 2018 | Markus Borg

V	tid	måndag 8	10	12	13	15	K	tisdag 8	10	12	13	15	K	onsdag 8	10	12	13	15	K	torsdag 8	10	12	13	15	K	fredag 8	10	12	13	15	K	lö	sö
12	Aktivitet	19/3	F1 MA1														Lab1			Ö1													
	Grupp 1-4																Alfa	PW		3308													
	Grupp 5-8																Falk	PW		2116					L1								
	Grupp 9-12																	Alfa		PW	3308												
	Grupp 13-16																	Beta		PW	2116												
13	Aktivitet	26/3	F2 MA1														Lab2			Ö2						Långfredag							
	Grupp 1-4																Alfa			3308													
	Grupp 5-8																Falk			2116													
	Grupp 9-12																	Gamma			3308												
	Grupp 13-16																	Falk			2116												
14	Aktivitet	Annandag påsk						Tentamensperiod						Tentamensperiod						Tentamensperiod						Tentamensperiod							
	Grupp 1-16																																
15	Aktivitet	Tentamensperiod						Tentamensperiod						Tentamensperiod						Tentamensperiod					L2	Tentamensperiod							
	Grupp 1-16																																
16	Aktivitet	16/4	F3 MA1														Lab3			Ö3													
	Grupp 1-4																Alfa			3308													
	Grupp 5-8			L3									L4				Falk			2116													
	Grupp 9-12																	Alfa			3308												
	Grupp 13-16																	Gamma			2116												
17	Aktivitet	23/4	F4 MA1														Lab4			Ö4													
	Grupp 1-4																Alfa			3308													
	Grupp 5-8																Falk			2116													
	Grupp 9-12																	Gamma			3308												
	Grupp 13-16																	Alfa			2116												
18	Aktivitet	Siste april						1 maj									Kod-jour																
	Grupp 1-16																																
19	Aktivitet	7/5	F5 MA1																	Kristi himmelsfärdsdag											L6		
	Grupp 1-16																																
20	Aktivitet	14/5	F6 MA4										L7												L8	Lundakarneval							
	Grupp 1-16																																
21	Aktivitet	21/5	F7 V:A																														
	Grupp 1-16																																
22	Aktivitet													Hemtenta																			
	Grupp 1-16																																
23	Återkoppling																																
24	Återkoppling																																
25	Återkoppling																									Midsommarafton							

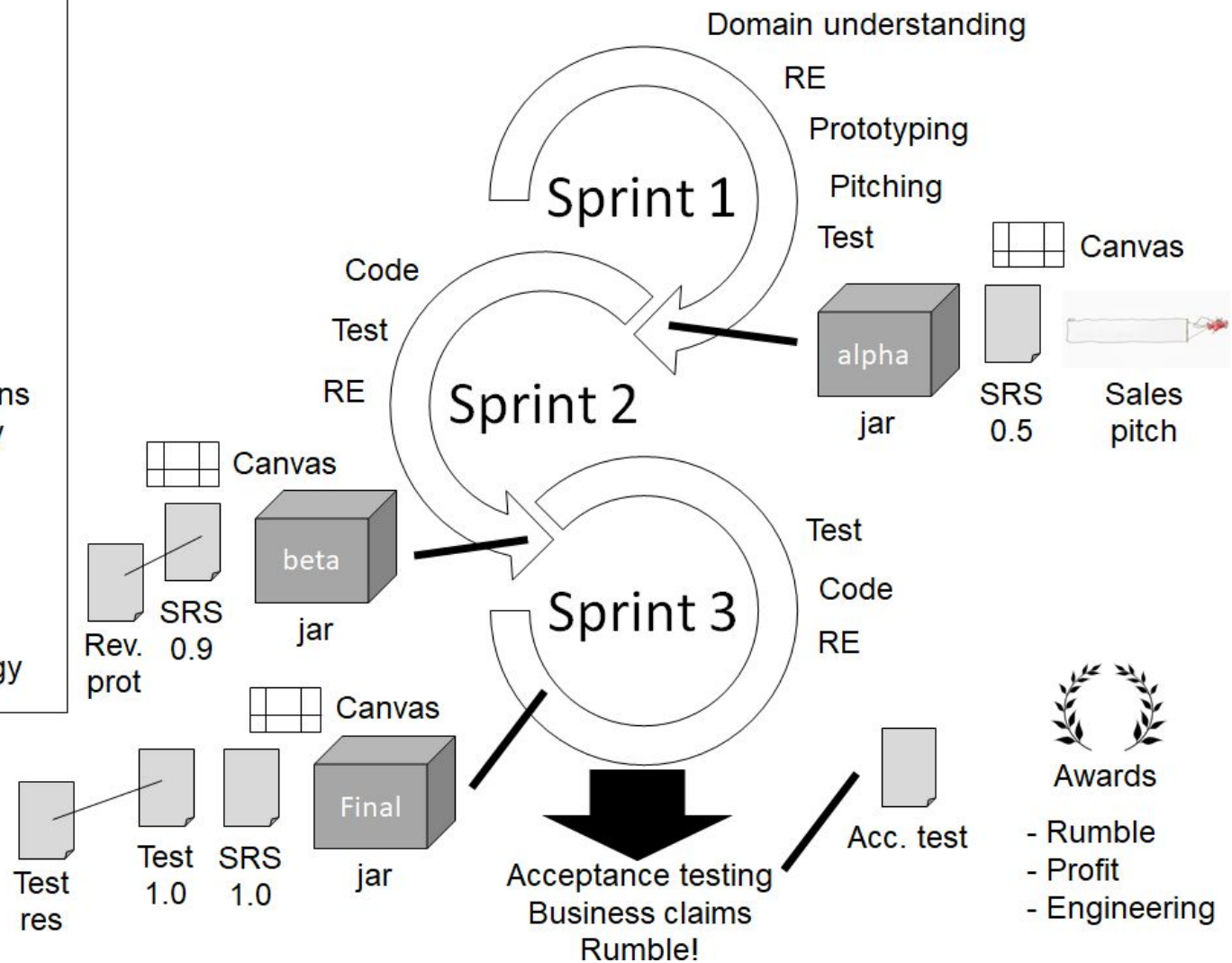


High-level goals

- Team formation
- Feature scoping
- Sales pitch

- Evolve product
- Maintain business relations
- Develop Rumble strategy

- Complete product
- High-volume sales
- Optimize Rumble strategy

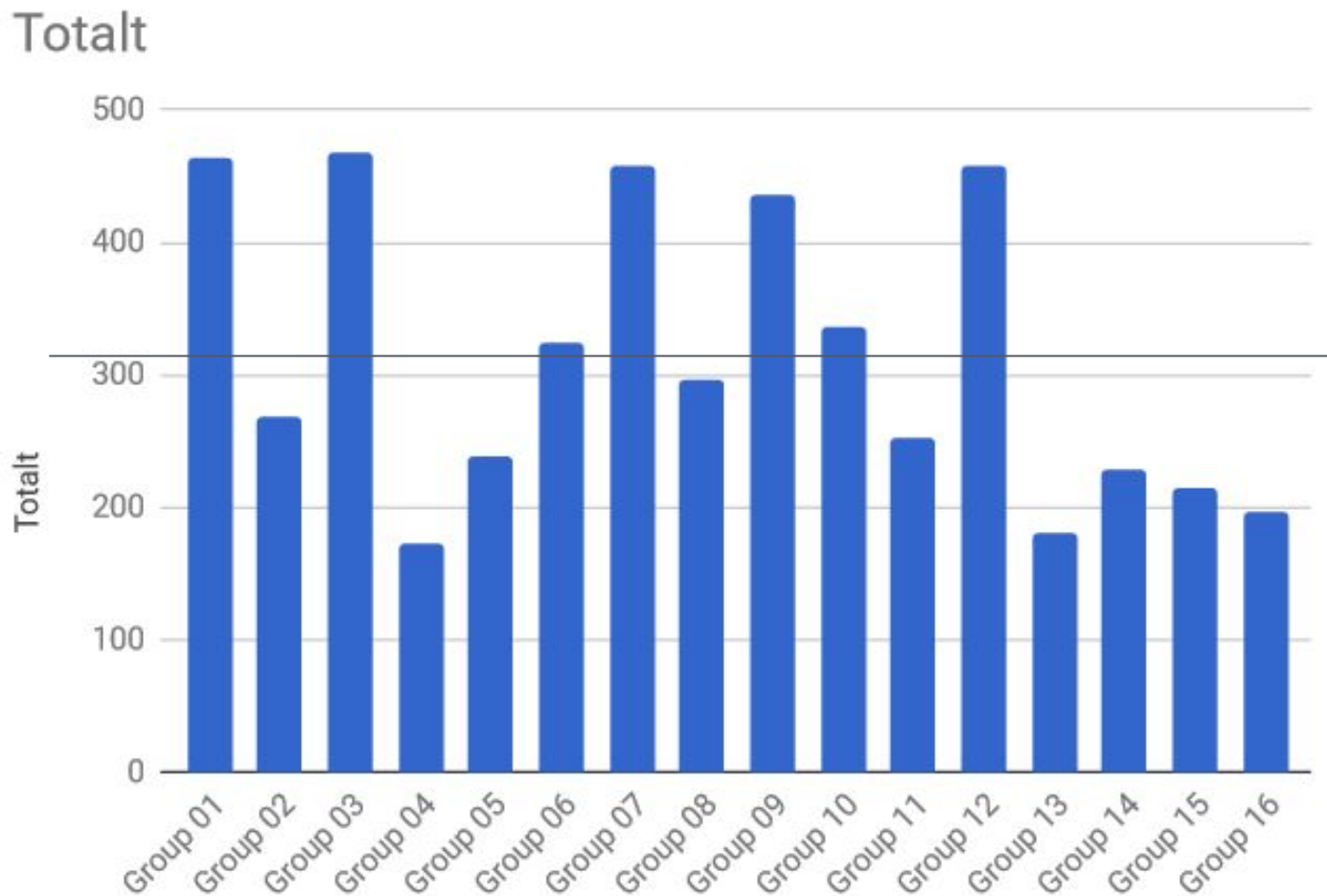


LUNDS UNIVERSITET

Det drar ihop sig för LU Rumble!

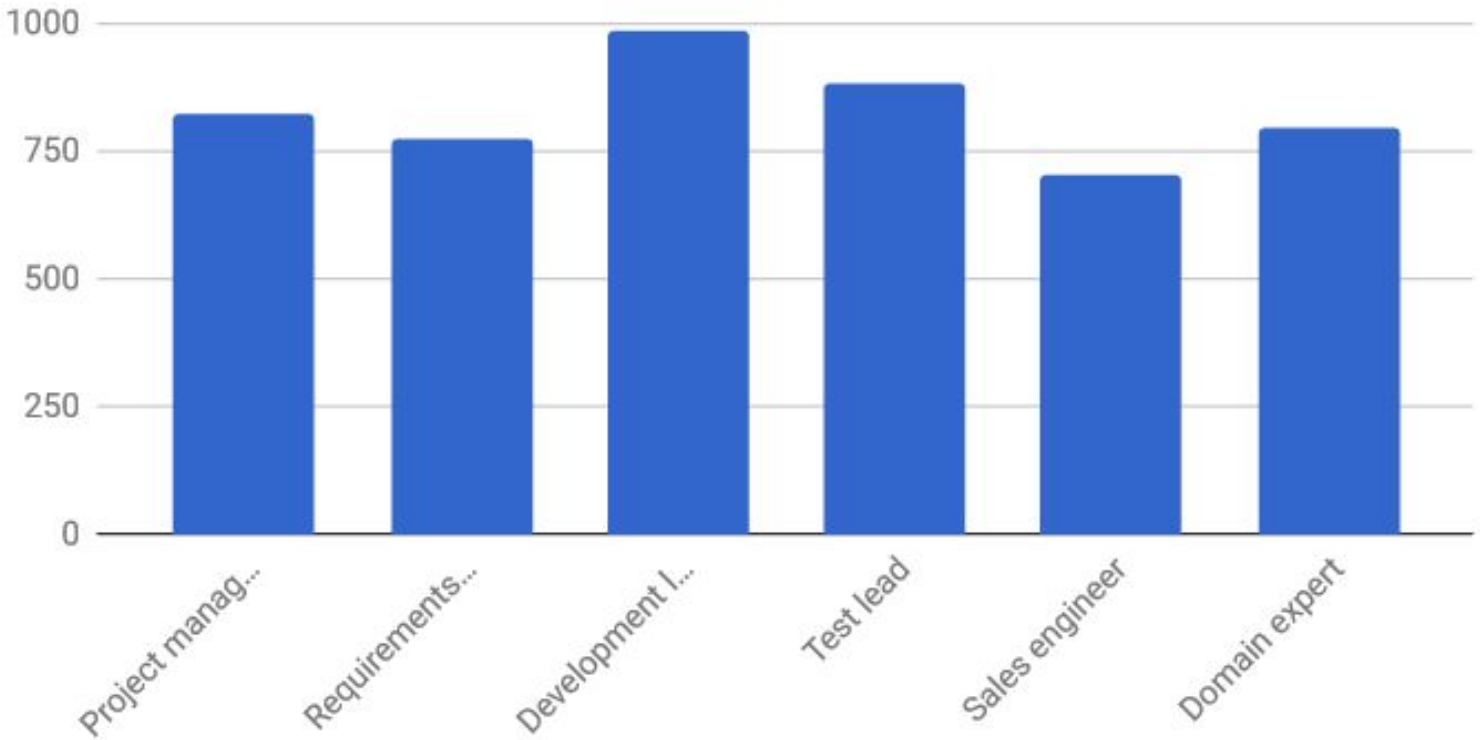
- Lagkonstruktion och lottning sker under helgen
- Välj slutgiltigt lagnamn!

Tidrapporter - Total tid per grupp



312 h

Tidrapporter - Total tid per roll

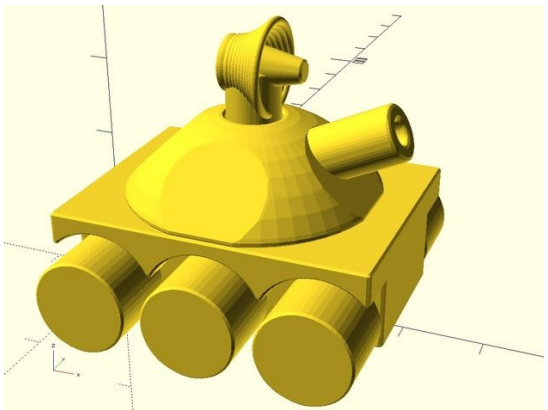


- Done!
- (Ledarutvecklare behöver sätta färg)

- Acceptanstestning avslutas imorgon (L7)
- Eventuellt business claims…
- Priser låsta Robot Market från L7

Engineering

Monetizing



Strategizing

Detta är på gång:

- Slutgiltig inköpsorder MAILAS till mig på torsdag 21.59 (L8)
- Meddela er ledarutvecklare!

L7: Tisdag kl 23.59

Acceptanstestrapport

- Maila till er handledare (+CC Markus) ihop med er leverantörs SRS
  - Valfritt rapportformat

Valfritt: Business claims (fordringar)

- Om skäl föreligger, mailas som ett separat dokument till er handledare
  - Valfritt format på fordringsbrev

L8: Torsdag kl 23.59

Slutgiltig inköpsorder

- En lista på de robotar ni köper för att komplettera er första
  - Specificera även vad ni anser att ni ska betala (jag kommer att dubbelkolla summan)
- Om ert köp leder till att ni besitter fler än fem robotar
  - Meddela vilka fem som ska tävla i LU Rumble

Lagfärgerna

- Meddela bums utvecklaren av er ledare

LUNDS UNIVERSITET

LUNDS UNIVERSITET

Vidareutveckling

Programvaruutveckling - Metodik 2018 | 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….

LUNDS UNIVERSITET

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

LUNDS UNIVERSITET

CARL QUIT. HE'S THE ONLY ONE WHO KNOWS HOW TO PROGRAM THE LEGACY SYSTEM.
www.dilbert.com
scottadams@aol.com
IT CAN'T BE THAT HARD. GO FIGURE IT OUT.
12-9-06
©2006 Scott Adams, Inc./Dist. by UFS, Inc.
?
FRACK.

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

LUNDS UNIVERSITET

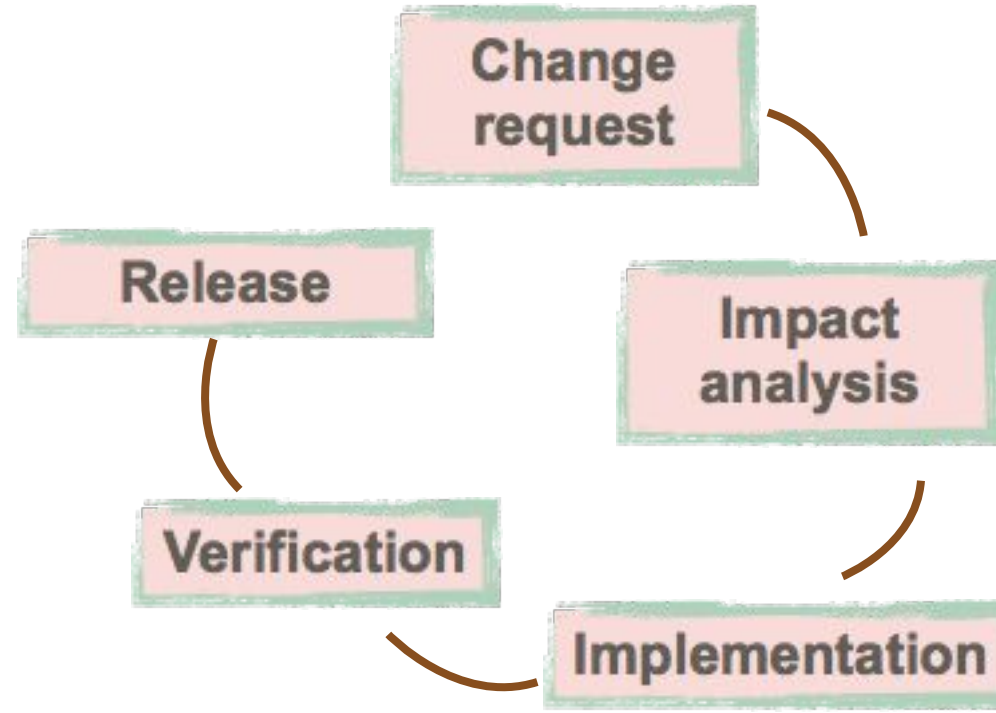
Underhåll efter release

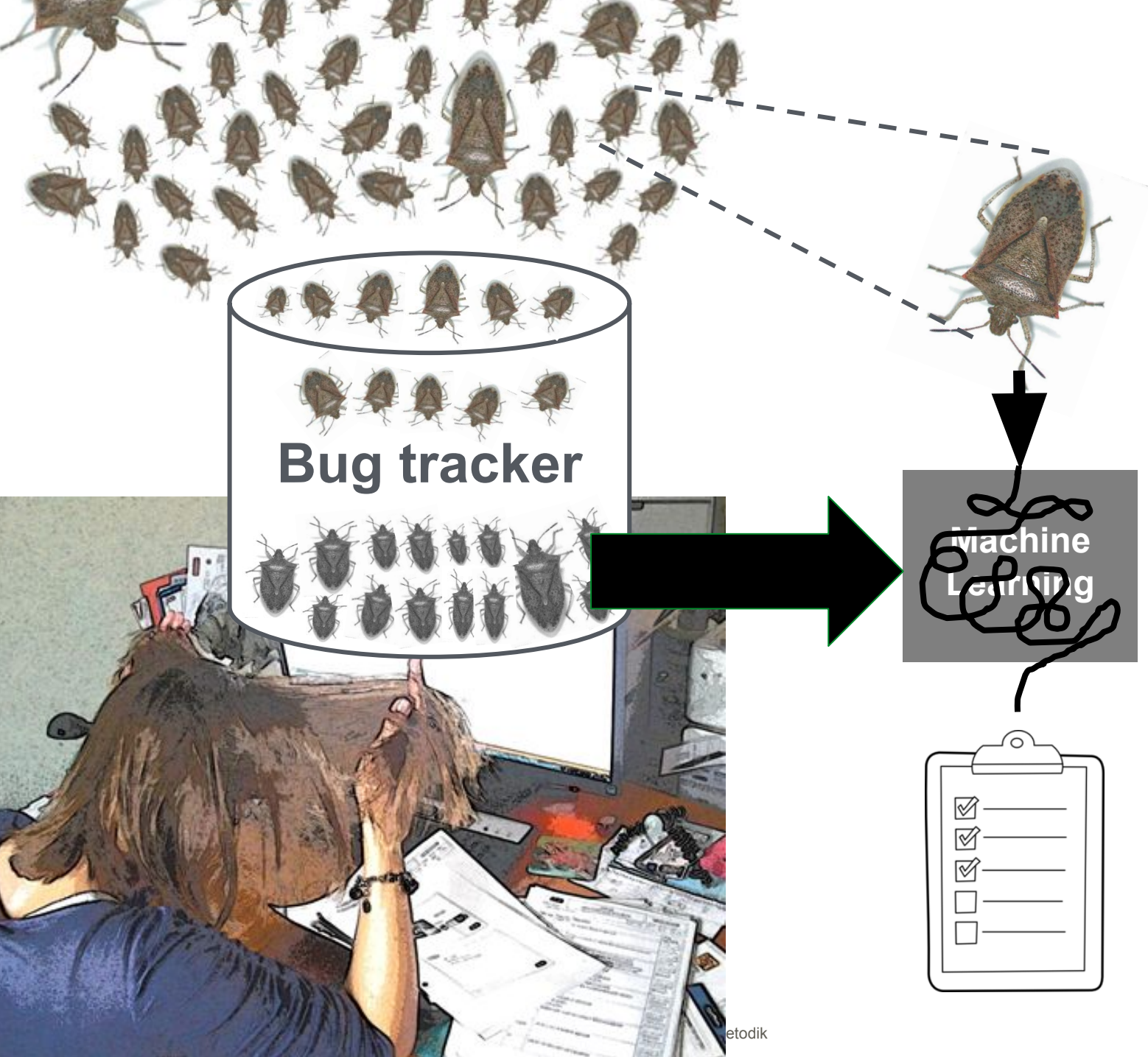
Kräver

- Ändringshantering
- Konfigurationshantering
- Påverkansanalys
- Spårbarhet
- Regressionstest

Man skiljer mellan:

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



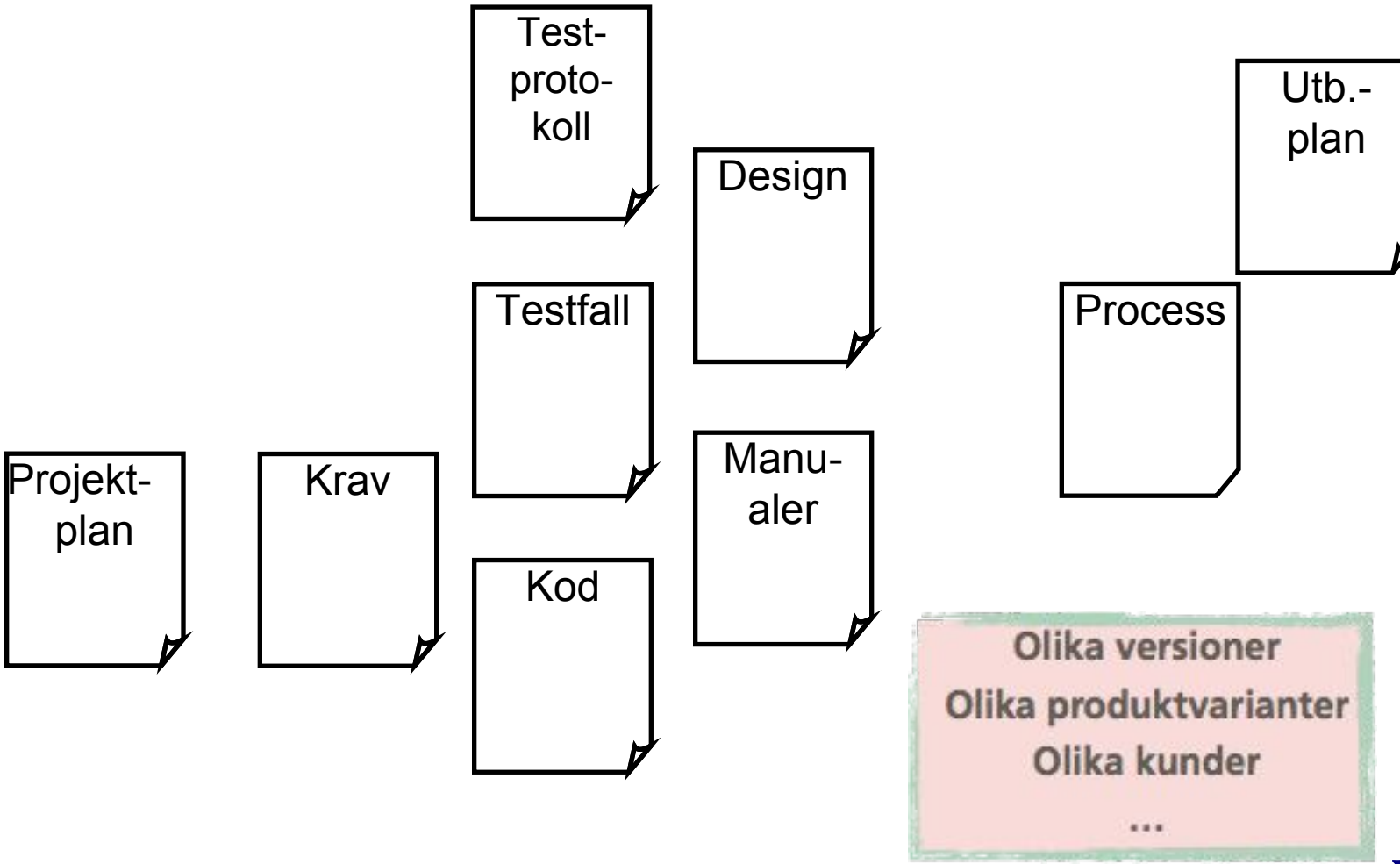

Bug tracker
Machine Learning
Vilka är viktigast?
Hur påverkar rättningen systemet?
Vem ska rätta den?

LUNDS UNIVERSITET

Konfigurationer och versioner

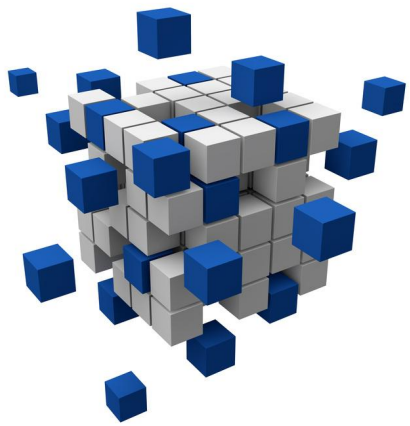
Programvaruutveckling - Metodik 2018 | Markus Borg

Många utvecklingsartefakter - olika versioner



Utmaningar i komplext informationslandskap

- Många utvecklare arbetar med artefakter som förändras
- Flera olika versioner behöver underhållas samtidigt
  - Olika releaser (Free version, Pro Version, Enterprise Edition etc.)
  - Skräddarsydda releaser åt särskilda kunder
  - Releaser för olika plattformar (Win10, MacOS)



Stort behov av koordinering!

Typiska frågor man kan vilja ha svar på

- Vilka kunder har installerat en viss version av systemet?
- Vilken hårdvara och OS krävs för en viss version?
- Vilka versioner har levererats av ett visst system?
- Vilka versioner påverkas om jag ändrar en viss komponent i systemet?
- Hur många olika ändringar håller man på att göra av ett visst system?
- Hur många rapporterade men ännu ej rättade fel finns det i ett system hos en viss kund?

LUNDS UNIVERSITET

Konfigurationshantering hanterar röran!

Konfigurationshantering = strukturerad hantering av utvecklingsartefakter under och efter utvecklingsprocessen

- **Version** av artefakter (0.9, 1.0, 1.1 etc.)
  - Oftast den senaste som är viktig
- **Baseline** = version av artefakt som är formellt accepterad som bindande
- **Konfiguration** = mängd artefakter av utpekade versioner
  - T.ex. slutleverans, alpha- och betareleaser
- **Variant** = alternativa artefakter som ska vidareutvecklas
  - T.ex. beräkningsbibliotek för desktop och för mobil

Systembygge

Bygga en fungerande version av hela systemet från de komponenter som finns, t ex

- Senaste versionerna av allt
- En viss version av en viss produkt
- En viss version av en viss produkt till en viss kund
- En viss version av en viss produkt till en viss plattform
- …

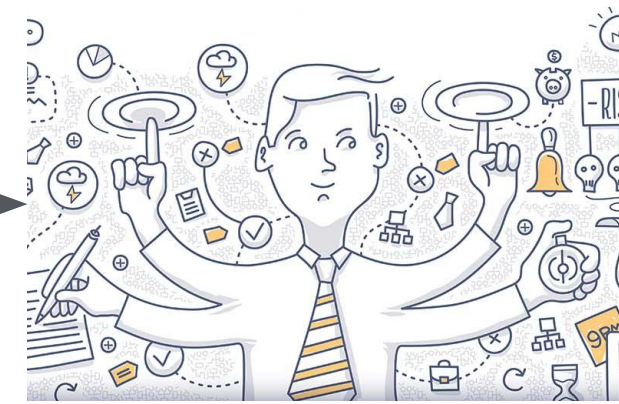
Behövs t.ex. vid systemtest och andra tester (och såklart vid leverans)

LUNDS UNIVERSITET

Några aktiviteter i konfigurationshantering

- Ändringshantering
  - Hantering, godkännande och uppföljning av ändringar
- Branch management
  - Planering och koordinering av interna utvecklingsgrenar
- Release-planering
  - Vilka konfigurationer ska släppas till kund? Vem ska få vad? (När och hur ska det ske? - i samråd med produktledningen)

software product management



Planering av konfigurationshantering

Återigen en avvägning i software engineering

- För tidigt införande skapar onödig byråkrati
- För sent införande leder till kaos

Vad ska konfigurationshanteras?

- Varje källkodsfil
- Kravspecifikationer
- Arkitekturmodeller
- Designmodeller
- Testspecifikationer
- Testkod
- Kanske användarmanualer?
- Kanske version av kompilatorn?
- Kanske hårdvara på testmaskinerna? Osv.

Samt givetvis:
- Vem?
- Hur?
- När?

Ändringshantering ≈ ordnad övergång

```
Begär ändring genom att fylla i formulär

Analysera ändringsbegäran

Om ändring nödvändig och korrekt {

    Utvärdera hur ändring kan göras + kostnad

    Skicka förfrågan till CCB

    Om ändring accepterad {

        Ändra i systemet

        Uppdatera ändringsbegäran

        tillverka ny version av systemet

    }

}
```

CCB – Change Control Board

Fattar strategiska beslut om vilka ändringar som ska ske

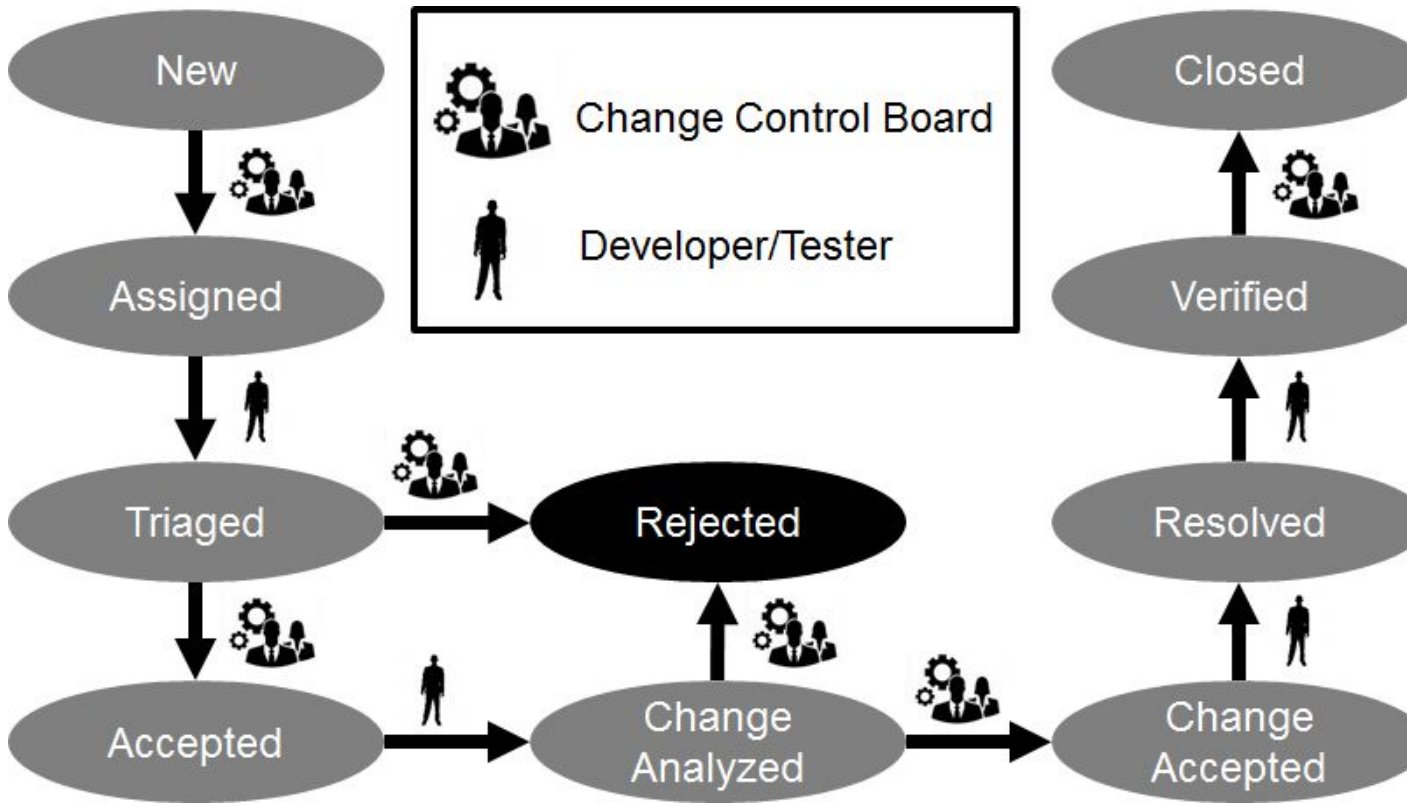
Inte bara tekniska beslut

Representerar kund, projekt och andra intressenter

För programvara som används i flera produkter blir det mer komplicerat

LUNDS UNIVERSITET

Ändringshantering på ABB



LUNDS UNIVERSITET

Produktledning

Programvaruutveckling - Metodik 2018 | Markus Borg

Plattformar inom produktutveckling

Komponenter delas mellan olika produkter i samma plattform

Black & Decker, t.ex.

- Battery Pack
- Motor Pack

Fordonsindustri, t.ex.

- Chassi
- Underrede
- Fjädring

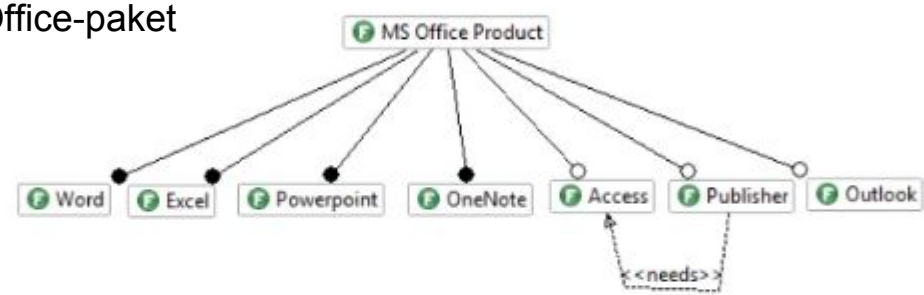
Software product line

"a **set** of software-intensive systems that share a **common, managed set of features** that satisfy the specific needs of a **particular market segment or mission** and that are **developed from a common set of core assets** in a prescribed way"

(Software Engineering Institute)

Konfigurera ut olika produkter

Exempel: Olika MS Office-paket

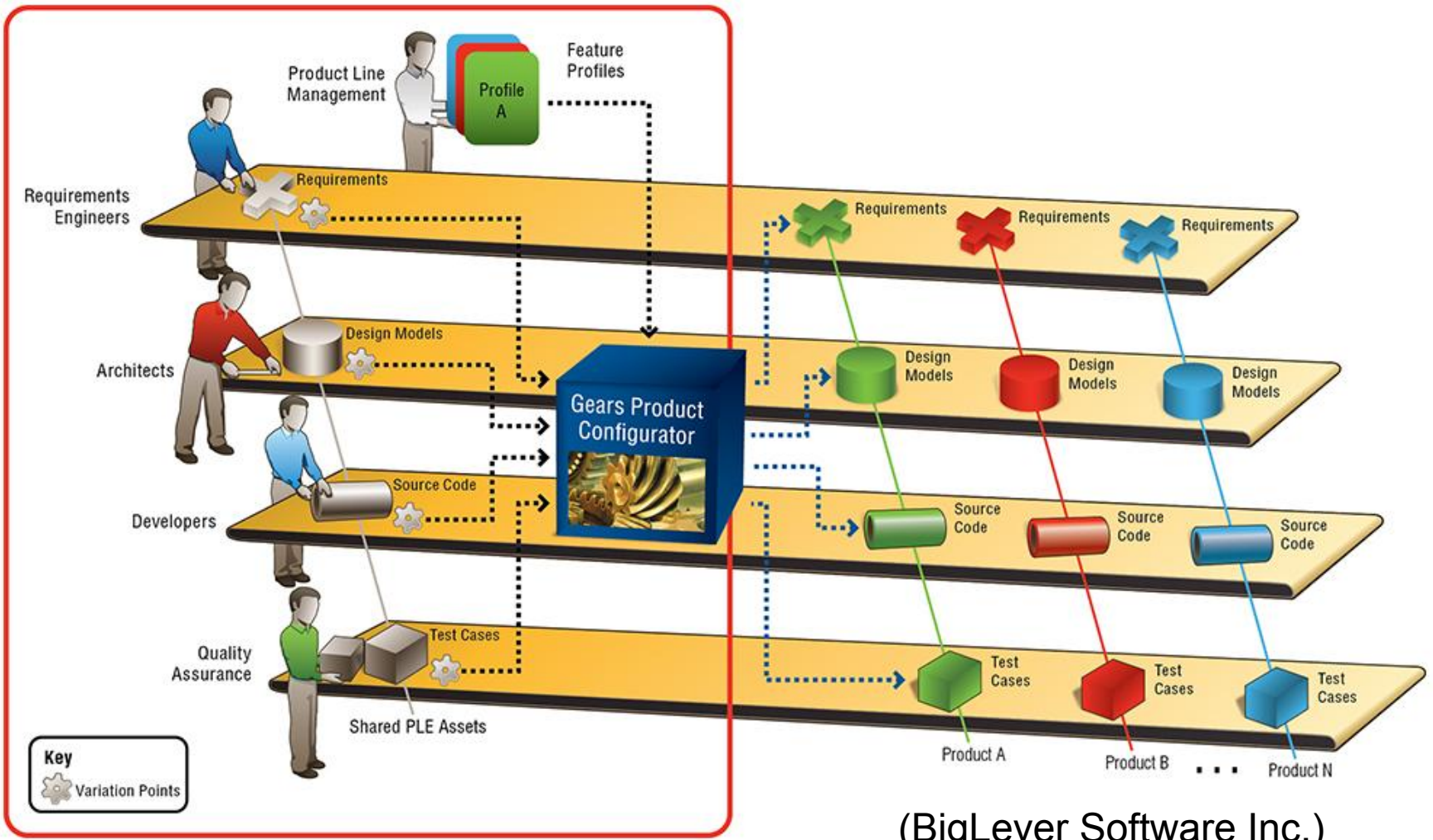


- C_1 = {Word, Excel, Powerpoint, OneNote}
- C_2 = {Word, Excel, Powerpoint, OneNote, Access}
- C_3 = {Word, Excel, Powerpoint, OneNote, Access, Publisher}
- C_4 = {Word, Excel, Powerpoint, OneNote, Outlook}
- C_5 = {Word, Excel, Powerpoint, OneNote, Access, Outlook}
- C_6 = {Word, Excel, Powerpoint, OneNote, Access, Publisher, Outlook}

Commonalities | Variabilities

(Jaime Chavarriaga,
Universidad de los Andes, Colombia)

Software product line



(BigLever Software Inc.)

Vad innebär produktlinjer?

Parallell utveckling innebär fördelar

- Lägre kostnader, fler produkter, kortare tid till marknad

Men även svårigheter

- Ökad komplexitet: t ex nya krav påverkar flera produkter
- Mer komplexa projektplaner, testning, organisation

Långsiktiga beslut för plattform, kortsiktiga beslut för produkter

Nya produkter innehåller funktioner från tidigare produkter

- stort sett aldrig specifikation “från 0”, dvs. mycket “arv”

Planering för produktlinjer

Produktlinjer innebär strategiskt återanvändande

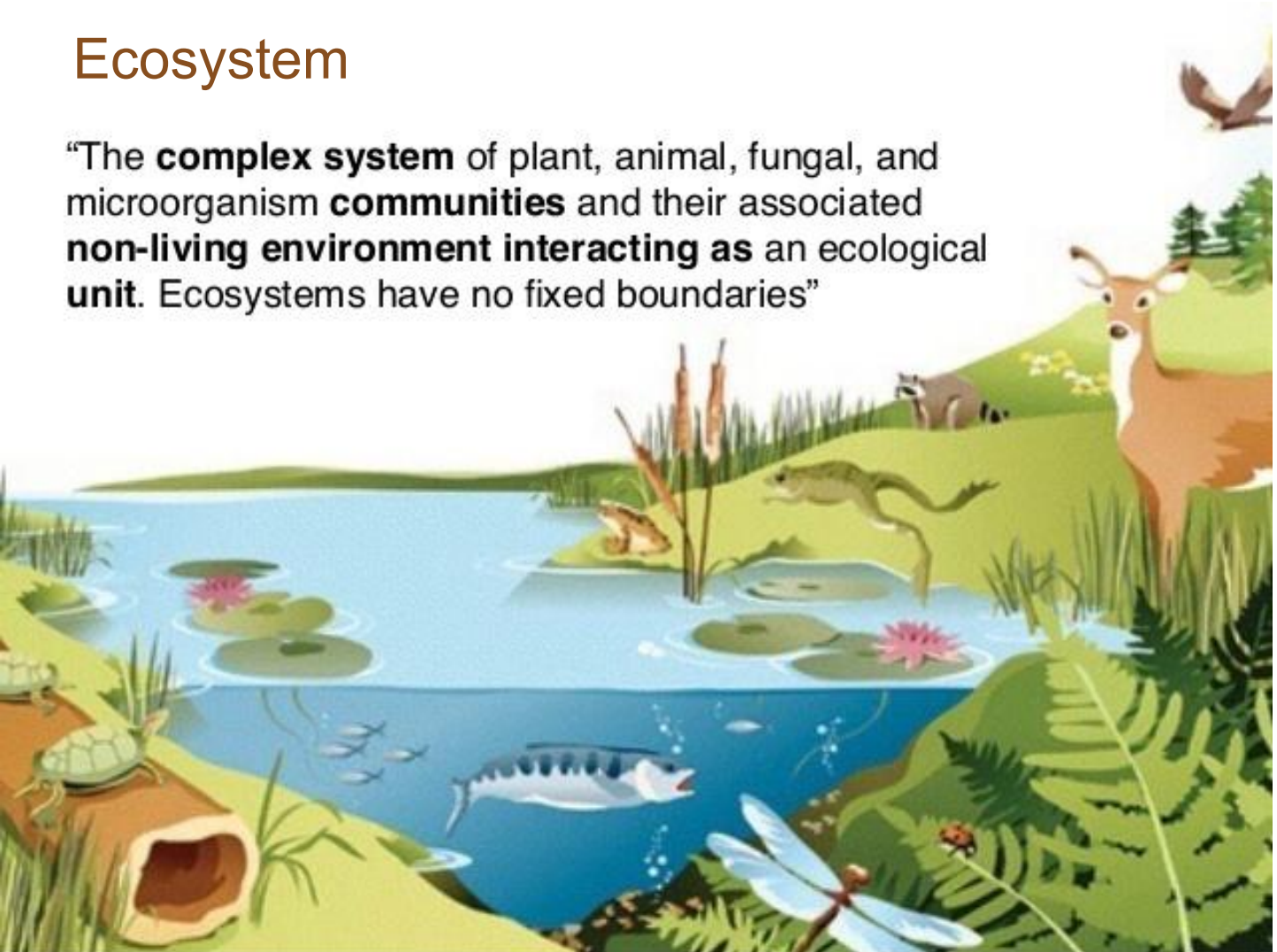
Identifiera likheter och variationspunkter

Grundförutsättning är att det finns (eller planeras) flera produkter med stora likheter

Nästa koncept kommer från naturen...

Ecosystem

"The **complex system** of plant, animal, fungal, and microorganism **communities** and their associated **non-living environment interacting as** an ecological **unit**. Ecosystems have no fixed boundaries"



(Tom Mens, Université de Mons)

LUND UNIVERSITY

Software ecosystems

Utveckling av programvarusystem är mer sällan stora kontraktsstyrda monolitiska system mellan två företag

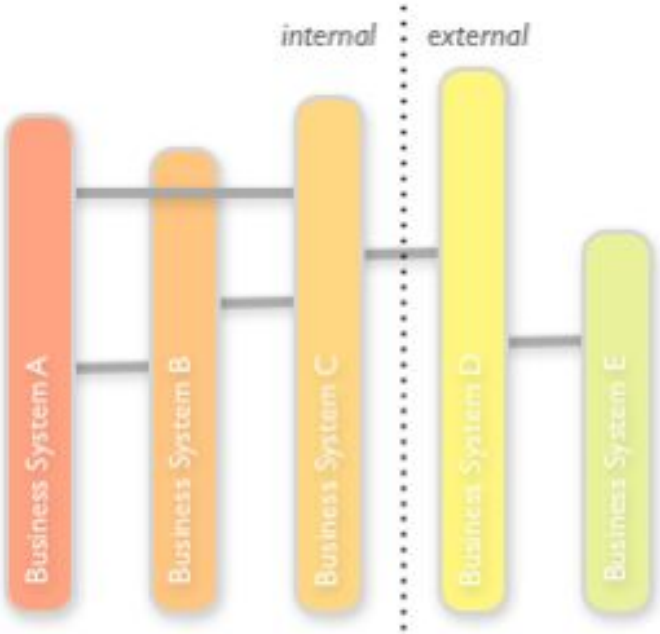
Istället:

"*a collection of software systems, which are developed and co-evolve in the same environment*" (Mircea Lungo)

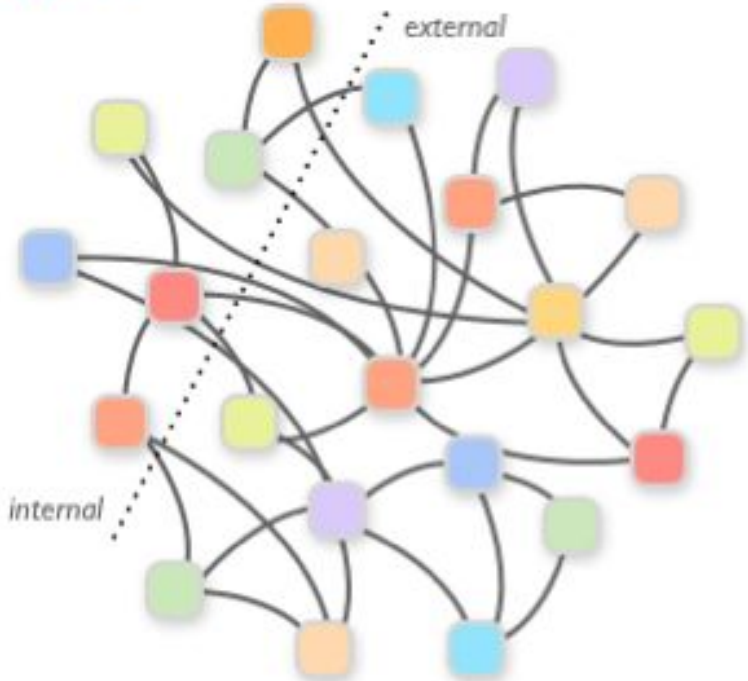
Exempel:

- inom företag
- app stores
- operativsystem
- programmeringsspråk

Moving from monolithic business stacks to networked open supply chains



VS.



Large, closed vertical systems tightly integrated with bespoke point-to-point connections

Small, open networked systems loosely integrated in a standards-based supply chain

creative commons Some Rights Reserved. 2011. DachisGroup by Dion Hinchcliffe

LUND UNIVERSITY

Vad innebär ekosystem?

Alla behöver inte utveckla allt

- Aktörer kan dra nytta av varandras utvecklingsinsatser

Öppna ekosystem kan stimulera innovation

- Aktörer kan fokusera på sina styrkor

Utmaningar kvarstår

- Någon behöver ta ansvar för gemensam plattform
- Aktörer måste komma överens om standarder
- Ansvar är svårt att reglera i kontrakt
- Svårt att genomföra release-planering

LUND UNIVERSITY

LUNDS UNIVERSITET

Snabbrepris och enkel utvärdering

Programvaruutveckling - Metodik 2018 | Markus Borg

Live-enkät

http://tiny.cc/etsa02-2018

Lab 1: Kom igång med Robocode *

	0	1	2	3	4	5	
Deltog ej/Vill ej svara	○	○	○	○	○	○	Mycket givande!

Positivt/negativt/nya idéer?

Long answer text

Snabbrepris och utvärdering

1. Teoretiska moment, i kronologisk ordning
   - Föreläsningar
   - Laborationer
   - Övningar

2. Praktiska färdigheter

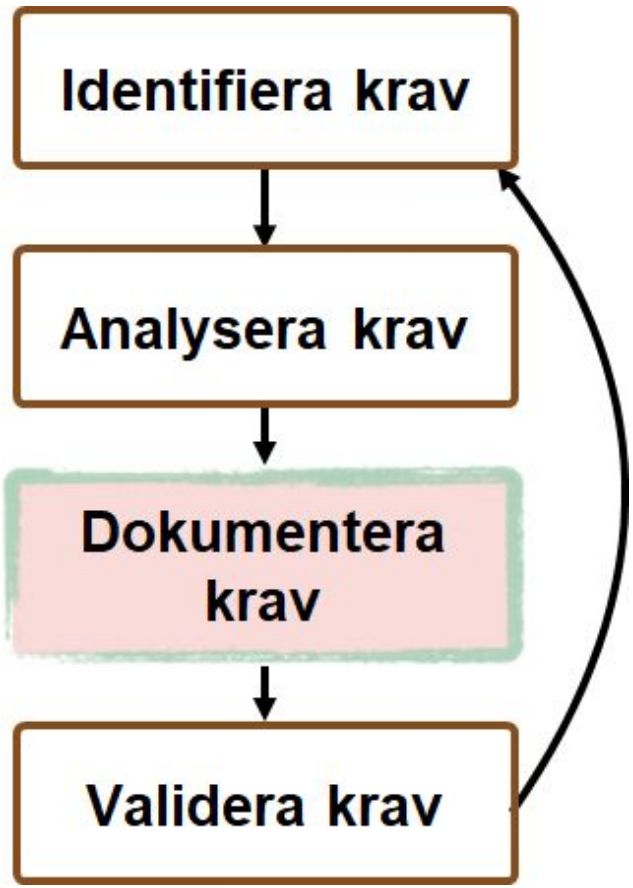
LUNDS UNIVERSITET

Teoretiska moment

Programvaruutveckling - Metodik 2018 | Markus Borg

F1: Intro, gruppindelning, kravhantering

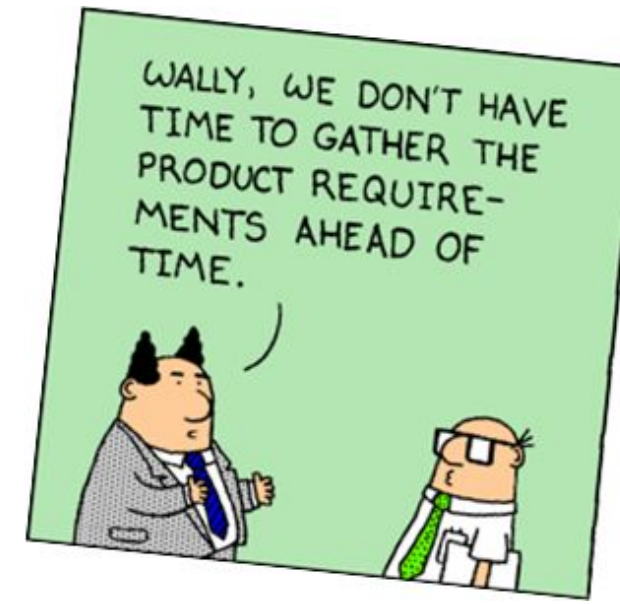
Grundbult i lyckade mjukvaruprojekt!



- Funktionella krav
- Kvalitetskrav

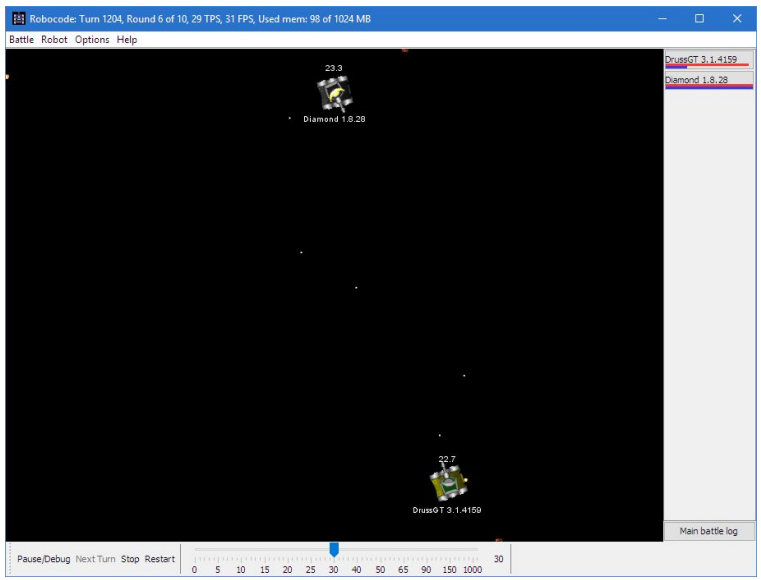
- Korrekt, komplett, otvetydigt, verifierbart etc.

- Skall-krav
- Användningsfall



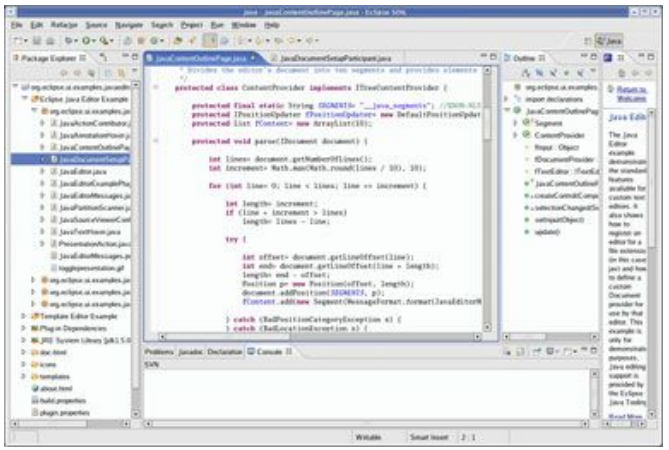
Lab 1: Kom igång med Robocode

- Grundläggande Robocode-tutorial
- Utveckla en första robot
- Robocode och Eclipse



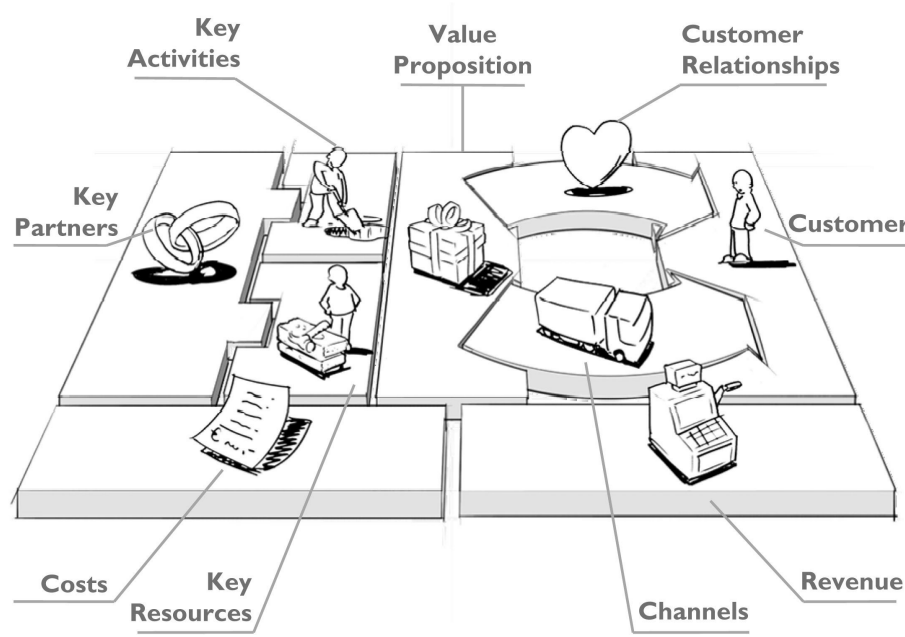
Path to robot src

Java Build Path



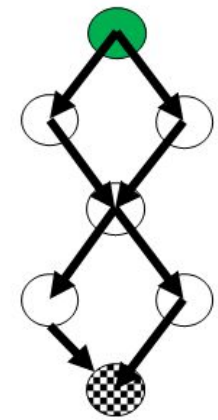
Ö1: Affärsmål, produktmål, features

- Marknadsdriven kravhantering
- Feature scoping
- Lean canvas
  - Affärsplan på 1 sida



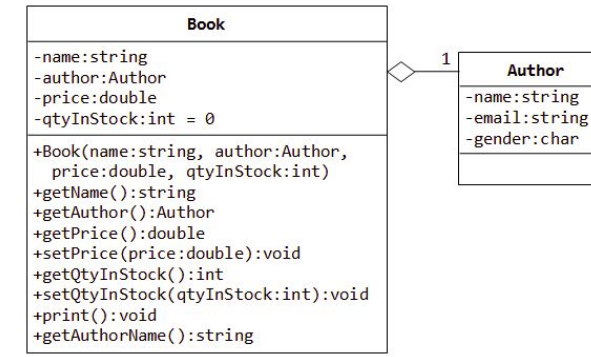
F2 del 1: Testning och whitebox-tekniker

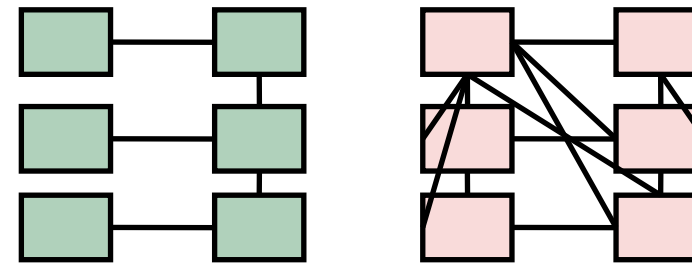
- Verifiering och validering
- Dynamisk och statisk testning
- Nivåer av testning
  - Acceptanstestning
  - Systemtestning
  - Integrationstestning
  - Enhetstestning
- Kodtäckning



F2 del 2: Objektorienterad design

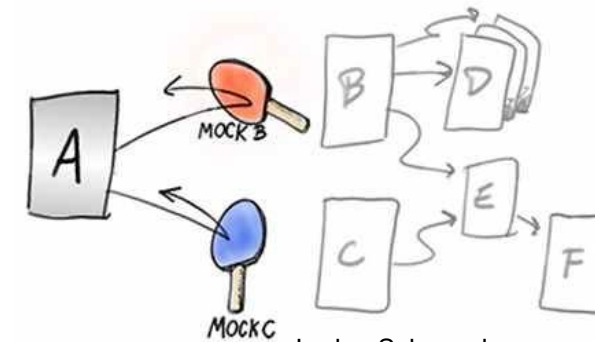
- Design av komponenter på klassnivå
- Unified Modeling Language (UML)
- Koppling och sammanhållning
- Designmönster



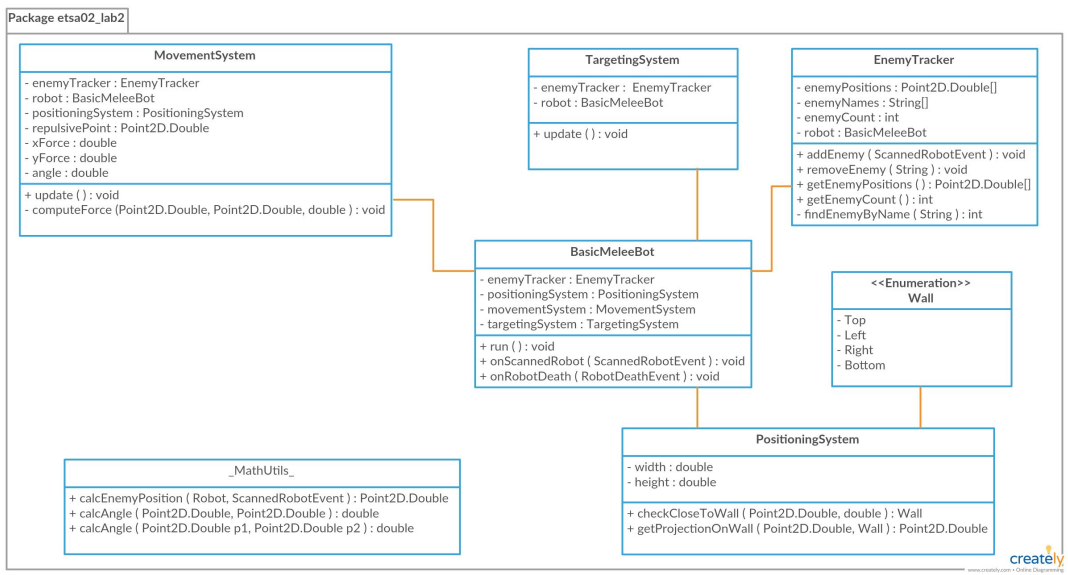


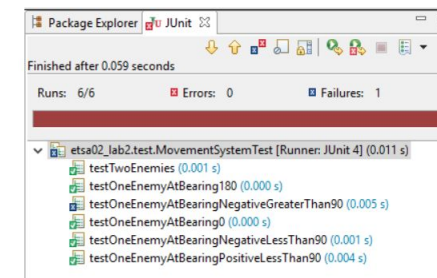
Lab 2: Refaktorisering och JUnit

- Skapa projekt i Eclipse, importera källkod
- Refaktorisera design
- Exekvera enhetstester med JUnit
- Mockobjekt för att isolera enheter



Jordan Schaenzle (2012)





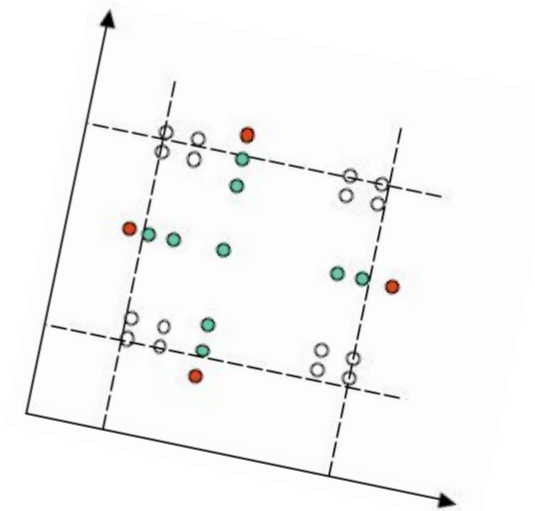
Ö2: Marknadskommunikation

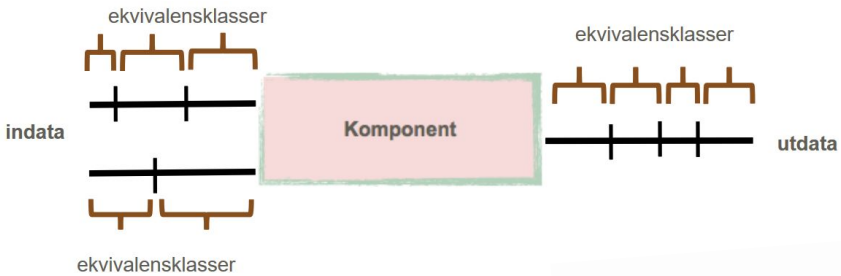
- Marknadsföring
- Sales promotion
- Sales pitch
- Storyboarding

CC 3.0 Flickr:tmray02

F3 del 1: Blackbox testing

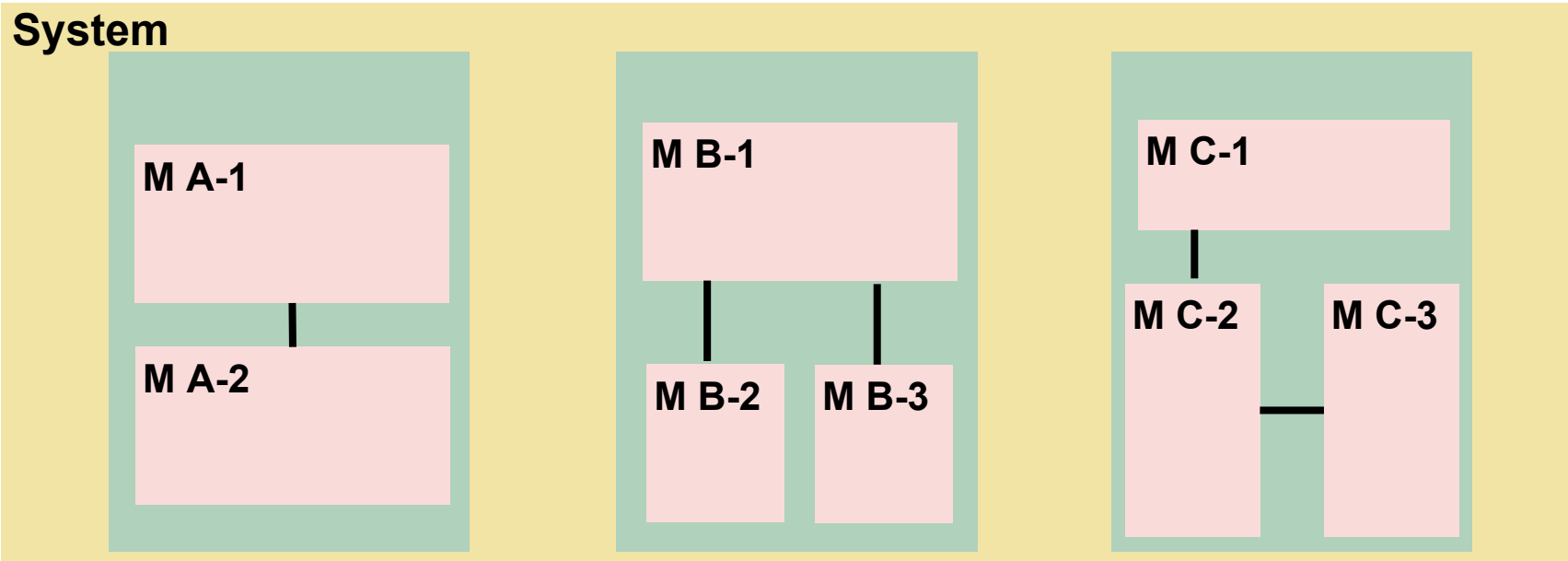
- Ekvivalenspartitionering
- Gränsvärdestestning
- Parvis testning
- När har man testat färdigt?





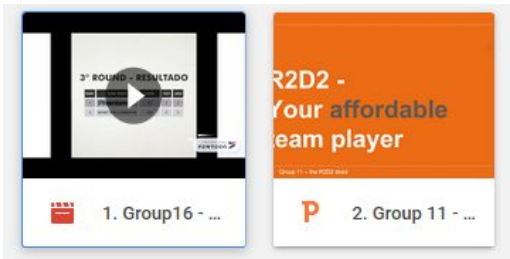
F3 del 2: Arktikturdesign

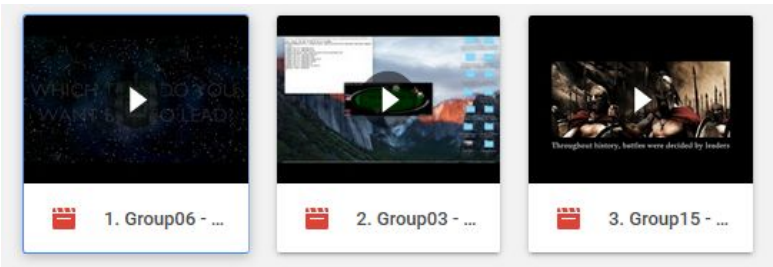
- Nedbrytning av systemets övergripande struktur
  - System, sybsystem, moduler, komponenter
- Länk mellan krav och detaljerad design
- Styrs ofta av kvalitetskraven

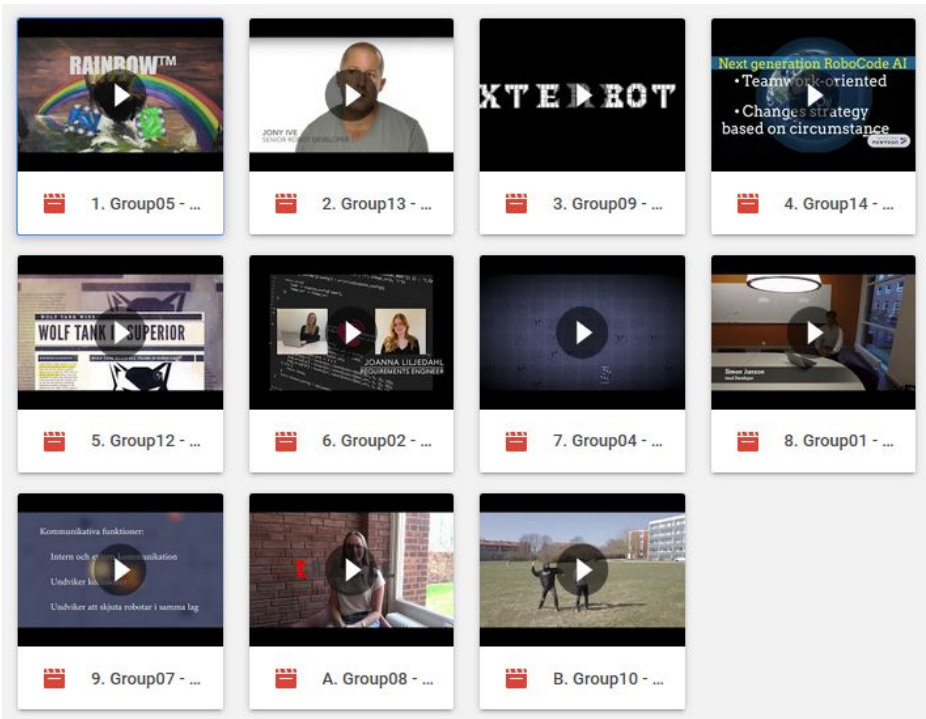


F3 del 3: Robotmässa

- Filmvisning
  - 16 stycken video pitches (max 2 min)

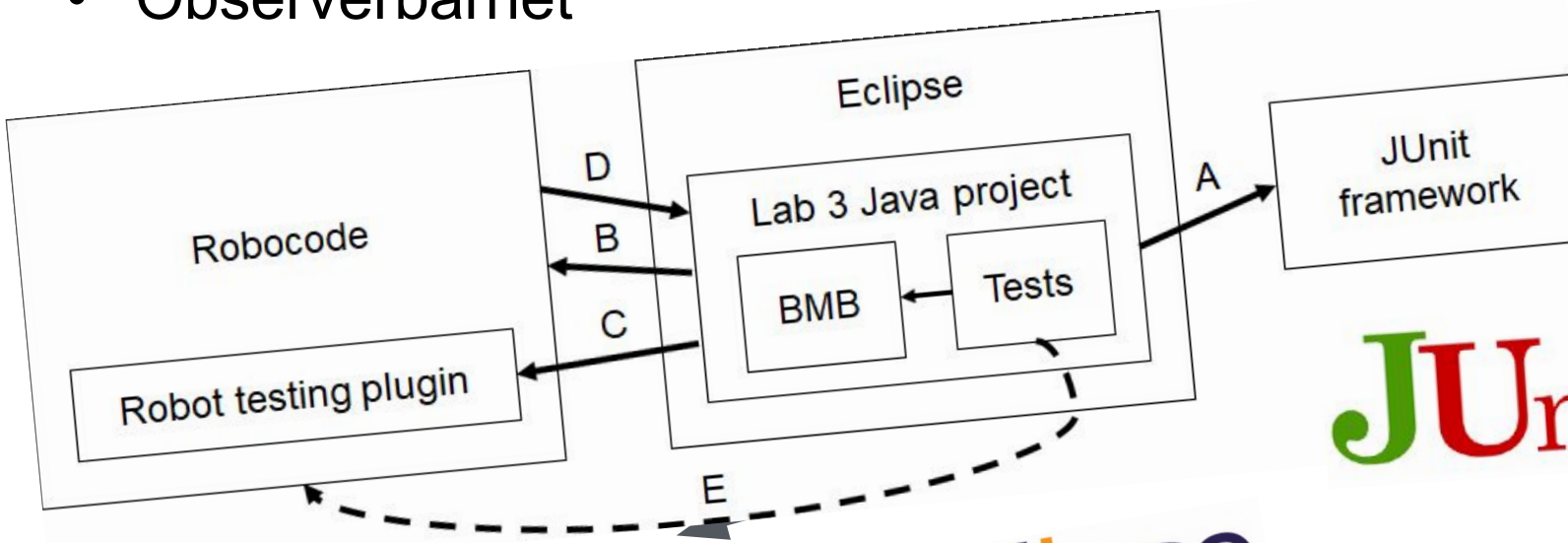






Lab 3: Systemtestning med RobotTestBed

- Automatiserade systemtester
- Test av funktionella krav och kvalitetskrav
- Observerbarhet

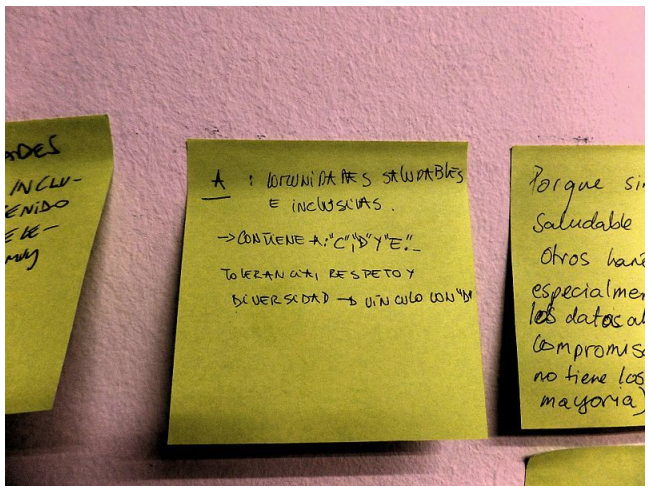


Ö3: Krav-workshop, detaljerade krav

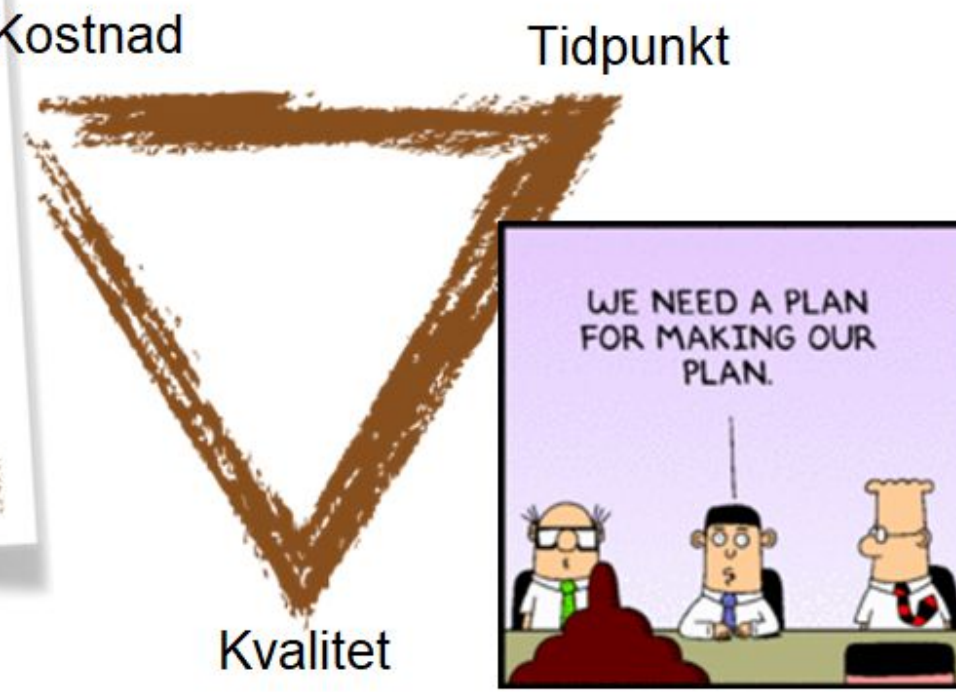
- Syftet med krav
- Egenskaper hos bra krav
- Grupparbete kring detaljerade krav för en feature
- Peer feedback med post-its

SRS-B-42

The system shall support 100 simultaneous users.

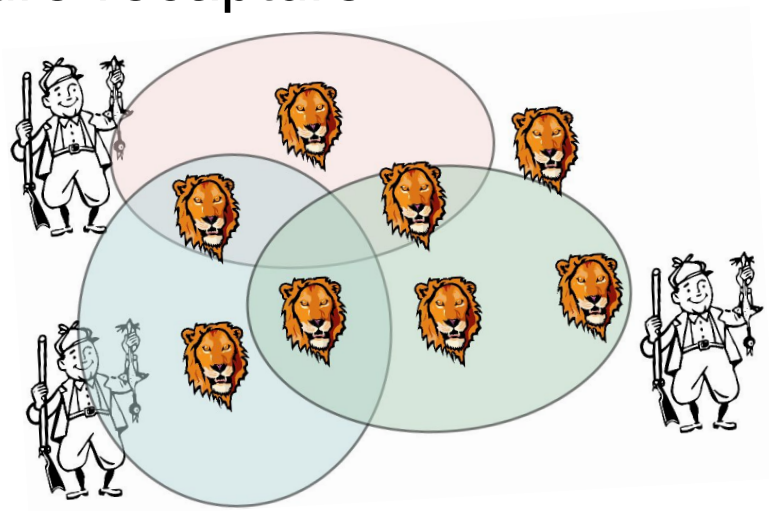


F4 del 1: Projektplanering och ledning



F4 del 2: Granskning

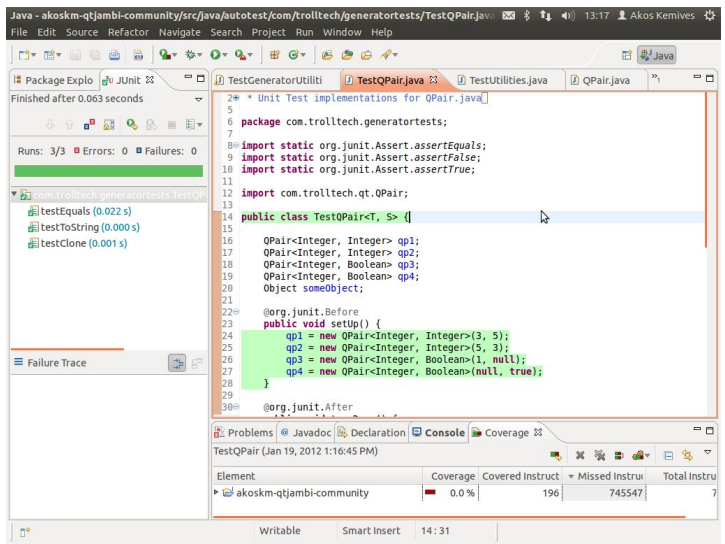
- Kodgranskning
- Fagan-inspektion
- Capture-recapture



Planering → **Intro -duktion** → **Individuell granskning** → **Gransk -ningsmöte** → **Omarbete** → **Uppföljning**

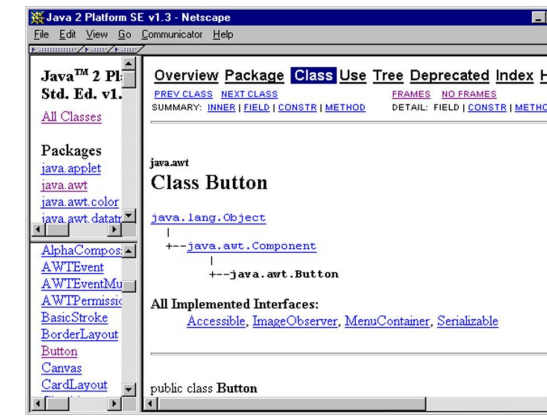
Lab 4: Kodtäckning, statisk kodanalys, Javadoc

- Verktygsstöd för utvecklare
  - EclEmma för kodtäckning
  - SonarLint och SpotBugs för statisk kodanalys
  - Javadoc för dokumentation



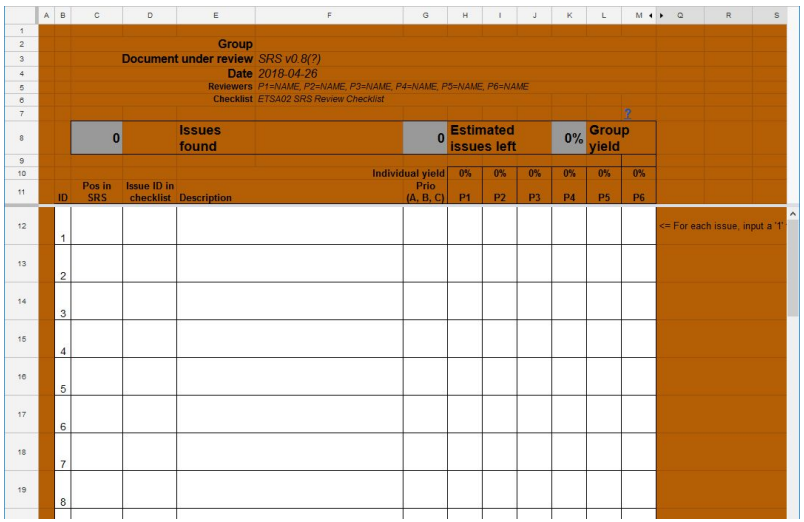
sonarlint

SpotBugs



Ö4: Individuell granskning, granskningsmöte

- Sex steg i Fagan-inspektion
- Individuell granskning av kravspecifikation
- Formellt granskningsmöte
- Granskningsprotokoll med capture-recapture

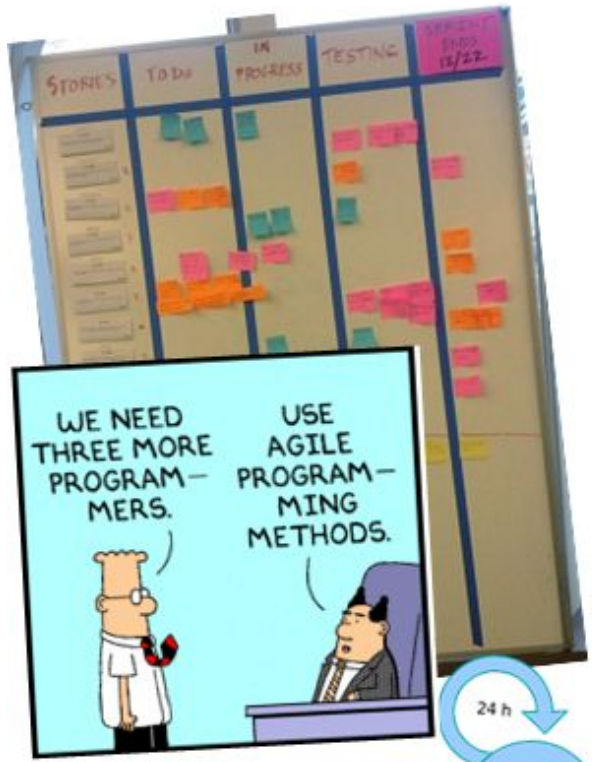


Group	
Document under review	SRS v0.8(?)
Date	2018-04-26
Reviewers	P1=NAME, P2=NAME, P3=NAME, P4=NAME, P5=NAME, P6=NAME
Checklist	ETSA02 SRS Review Checklist

0	Issues found	0	Estimated issues left	0%	Group yield

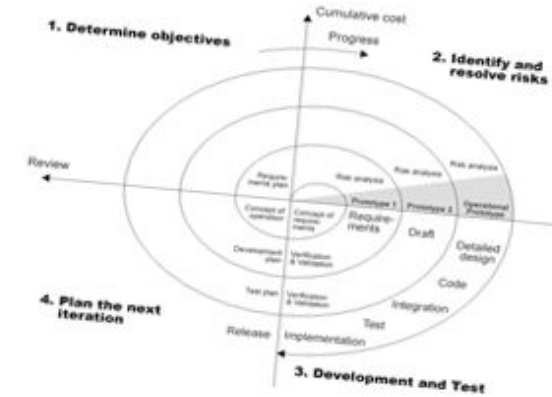
	ID	Pos in SRS	Issue ID in checklist	Description	Prio (A, B, C)	P1	P2	P3	P4	P5	P6	
Individual yield						0%	0%	0%	0%	0%	0%	
	1											<= For each issue, input a "1"
	2											
	3											
	4											
	5											
	6											
	7											
	8											

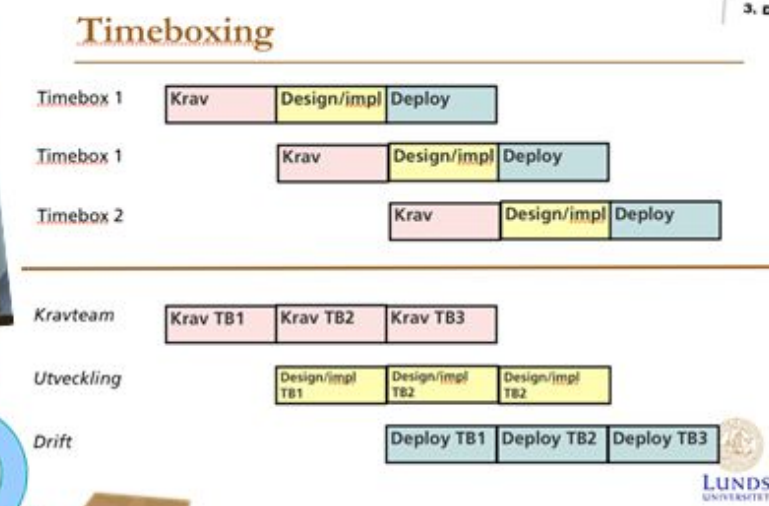
F5: Utvecklingsprocesser

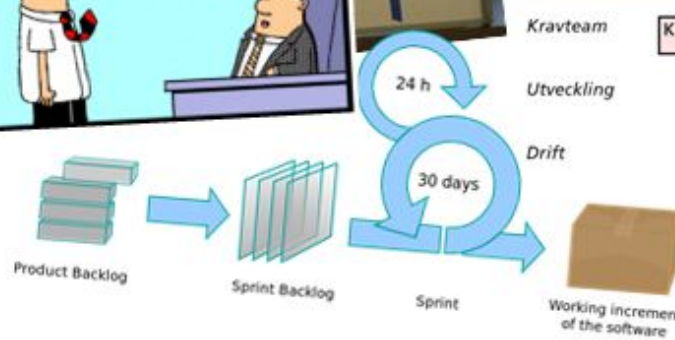


Processmodeller

- Linjära
- Evolutionära
- Iterativa
- Lättrörliga



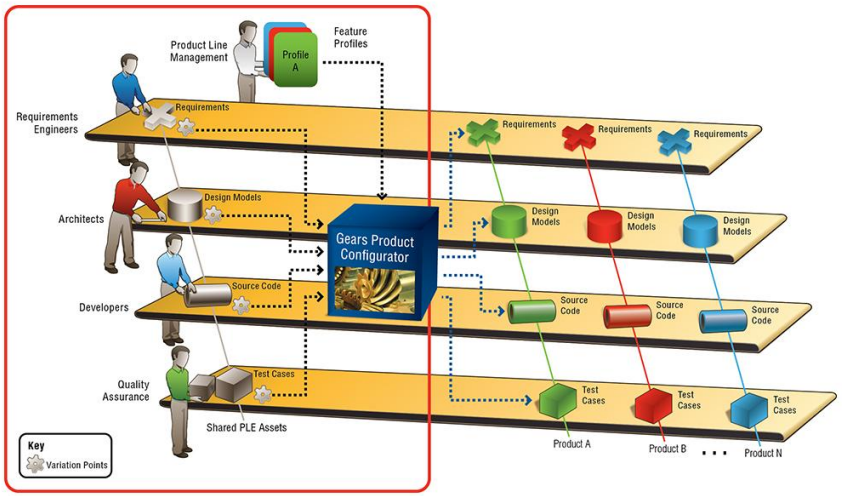




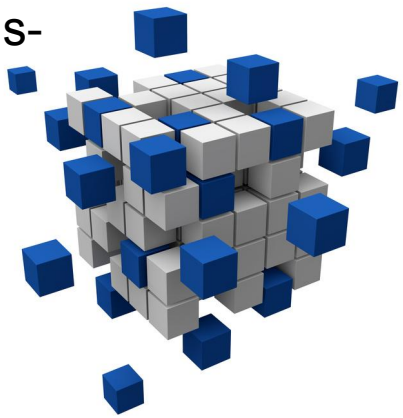
F6: Vidareutveckling, produktledning

Legacy systems

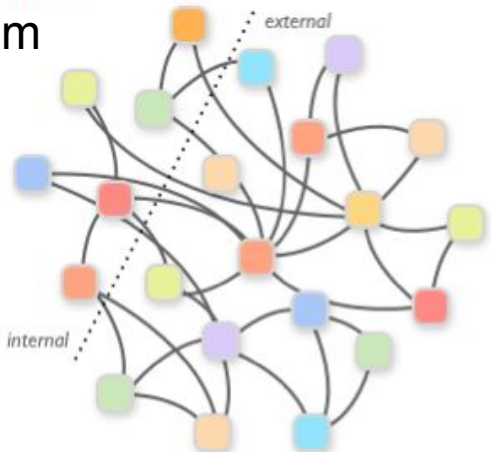
Produktlinjer



Konfigurations-hantering



Ekosystem



LUNDS UNIVERSITET

Praktiska färdigheter

Programvaruutveckling - Metodik 2018 | Markus Borg

Kravhantering

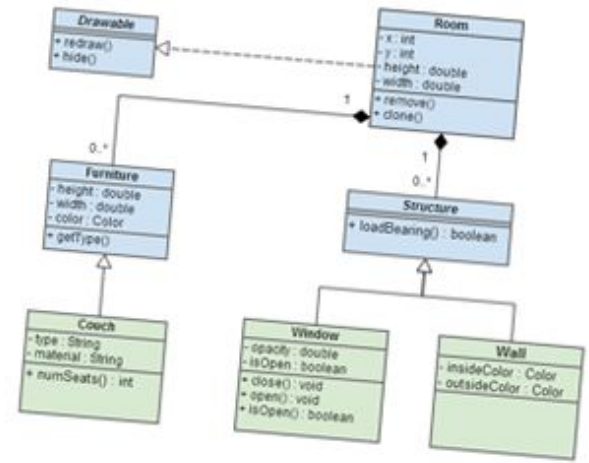
- Affärsmål
- Produktmål
- Features
- Kravspecifikation

Marknadsdriven programvaruutveckling

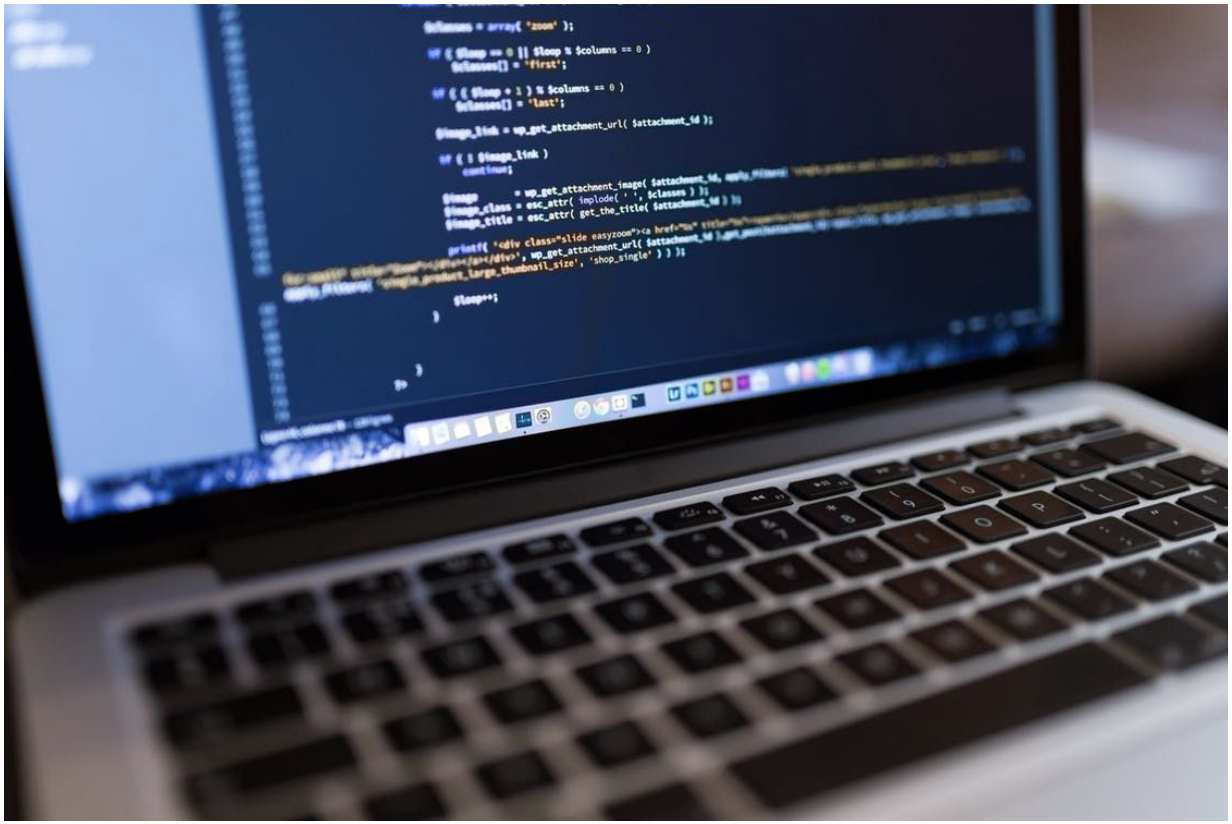
- Affärsplan med Lean canvas
- Nå ut på marknaden
  - Salespitch
  - Kommunikation med kunder
- Prissättning

Design

- Klassdiagram
- UML
- Refaktorisering

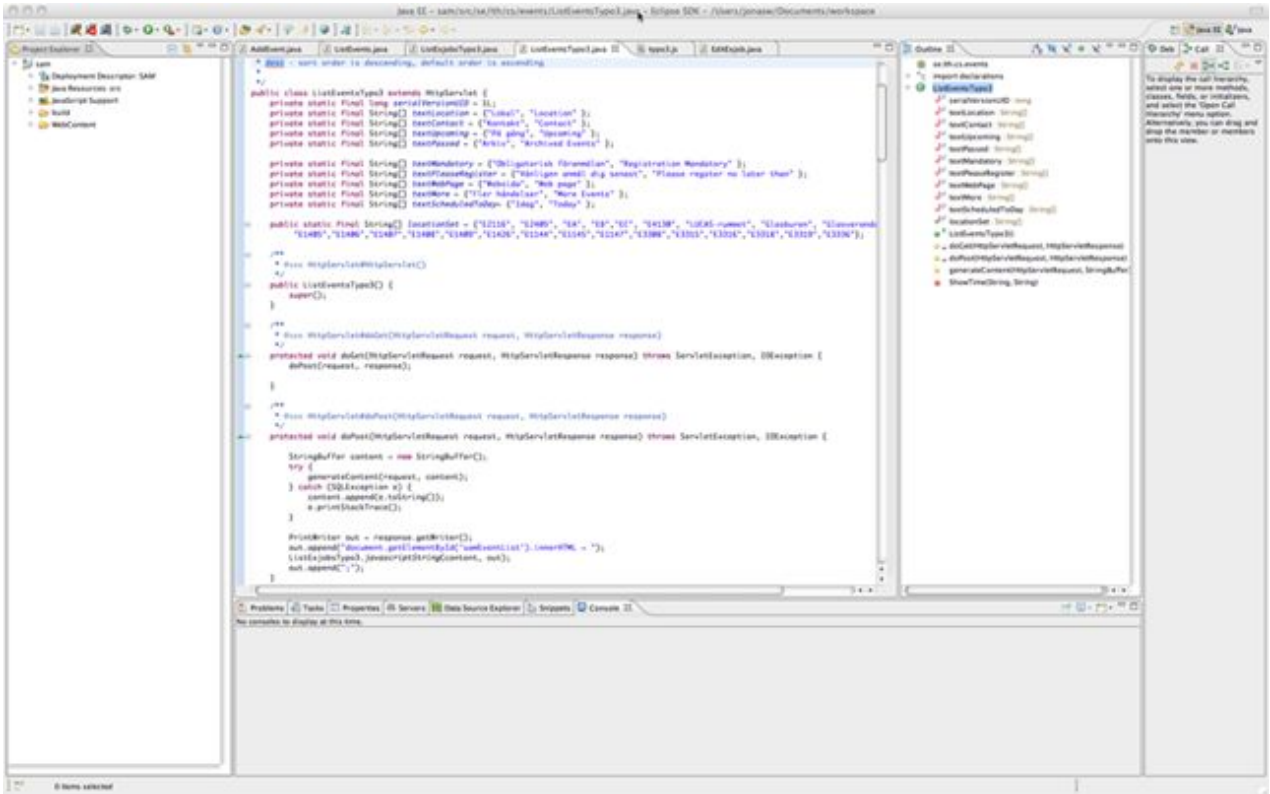


Programmering



Utvecklingsverktyg

- Eclipse som utvecklingens hemmaborg



Testning

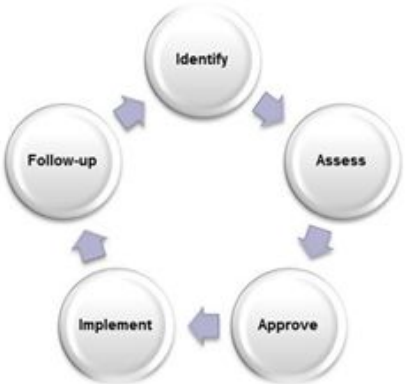
- Enhetstester
- Systemtester
  - Automatiska
  - Manuella
- Testspecifikation
- Testrapport

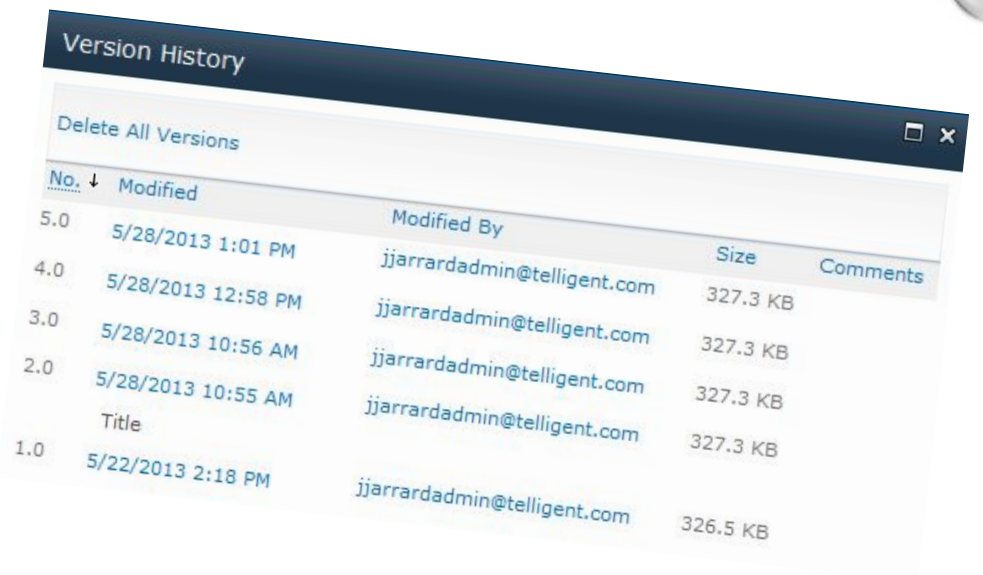
Granskningsarbete

- Granskningsprocessen
- Intern dokumentgranskning
- Kodgranskning

Versions- och konfigurationshantering

- git
- Versionshistorik för dokumentation
- Följa en releaseplan
  - Alpha (v0.5), Beta (v0.9), Final (v1.0)





Arbeta i team

- Kommunikation
- Planering
- Uppföljning

Teknisk dokumentation

- Arbeta med teknisk dokumentation
  - Planera
  - Skriva
  - Granska
  - Implementera återkoppling

LUNDS UNIVERSITET

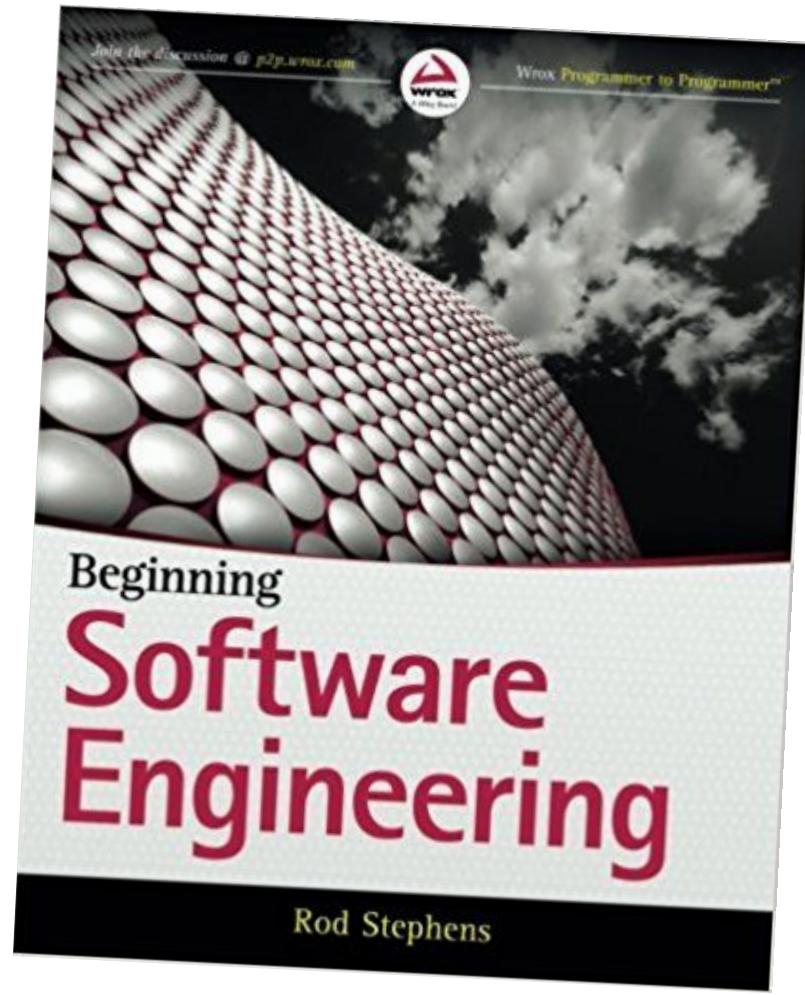
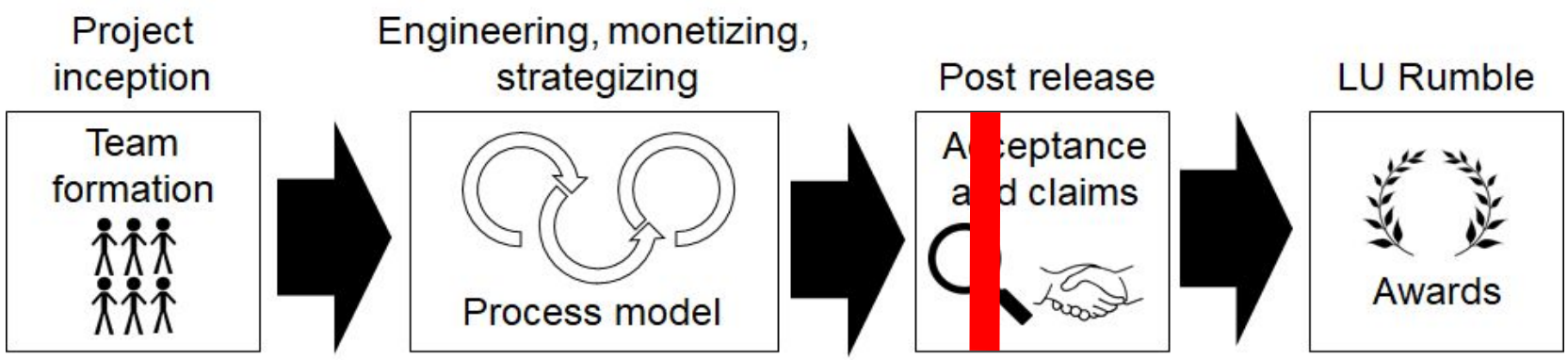
Slutligen

Programvaruutveckling - Metodik 2018 | Markus Borg

Robocode

Slack

Kursboken


Join the discussion @ p2p.wrox.com
Wrox Programmer to Programmer™
Beginning
Software
Engineering
Rod Stephens

V	tid	måndag 8	10	12	13	15	K	tisdag 8	10	12	13	15	K	onsdag 8	10	12	13	15	K	torsdag 8	10	12	13	15	K	fredag 8	10	12	13	15	K	lö	sö
12	Aktivitet	19/3	F1 MA1														Lab1			Ö1													
	Grupp 1-4																Alfa	PW		3308													
	Grupp 5-8																Falk	PW		2116					L1								
	Grupp 9-12																	Alfa		PW	3308												
	Grupp 13-16																	Beta		PW	2116												
13	Aktivitet	26/3	F2 MA1														Lab2			Ö2						Långfredag							
	Grupp 1-4																Alfa			3308													
	Grupp 5-8																Falk			2116													
	Grupp 9-12																	Gamma			3308												
	Grupp 13-16																	Falk			2116												
14	Aktivitet	Annandag påsk						Tentamensperiod						Tentamensperiod						Tentamensperiod						Tentamensperiod							
	Grupp 1-16																																
15	Aktivitet	Tentamensperiod						Tentamensperiod						Tentamensperiod						Tentamensperiod					L2	Tentamensperiod							
	Grupp 1-16																																
16	Aktivitet	16/4	F3 MA1														Lab3			Ö3													
	Grupp 1-4																Alfa			3308													
	Grupp 5-8			L3									L4				Falk			2116													
	Grupp 9-12																	Alfa			3308												
	Grupp 13-16																	Gamma			2116												
17	Aktivitet	23/4	F4 MA1														Lab4			Ö4													
	Grupp 1-4																Alfa			3308													
	Grupp 5-8																Falk			2116													
	Grupp 9-12																	Gamma			3308												
	Grupp 13-16																	Alfa			2116												
18	Aktivitet	Siste april						1 maj									Kod-jour																
	Grupp 1-16																																
19	Aktivitet	7/5	F5 MA1																	Kristi himmelsfärdsdag											L6		
	Grupp 1-16																																
20	Aktivitet	14/5	F6 MA4										L7												L8	Lundakarneval							
	Grupp 1-16																																
21	Aktivitet	21/5	F7 V:A																														
	Grupp 1-16																																
22	Aktivitet													Hemtenta																			
	Grupp 1-16																																
23	Återkoppling																																
24	Återkoppling																																
25	Återkoppling																									Midsommarafton