



LUNDS
UNIVERSITET

Föreläsning 4: Projektplanering & Granskning

ETSA02 Programvaruutveckling – Metodik 2018 | Markus Borg



Agenda F4

Projektplanering och projektplaner

Statisk testning - granskning

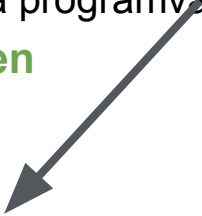
Om robotprojekten




LUNDS
UNIVERSITET

ETSA02 Formella kursmål

Kunskap och förståelse

- kunna definiera **grundläggande begrepp** inom utveckling av stora programvarusystem
 - kunna beskriva de **vanligaste processerna** för utveckling av stora programvarusystem
 - kunna förklara de viktigaste momenten i **kravhanteringsprocessen**
 - kunna förklara hur **testning** går till
 - kunna beskriva vad en **arkitekturdesign** är
 - kunna beskriva de viktigaste stegen i **projektplanering och projektuppföljning**
 - kunna beskriva hur organisationer planerar och genomför en **serie av projekt**
- 

Färdighet och förmåga

- kunna **utveckla projektplan**, **kravspecifikation** och **testplan** för ett mindre projekt
 - kunna **granska** projektplan, kravspecifikation och testplan för ett mindre projekt.
 - kunna skriftligen **formulera text i projektdokumentation**
- 

Värderingsförmåga och förhållningssätt

- förstå **komplexiteten** i uppgiften att utveckla ett programvarusystem.
- ha förståelse för **ingenjörens yrkesroll**



LUNDS
UNIVERSITET

Programvaruprojekt – Grundläggande begrepp

ETSA02 Programvaruutveckling – Metodik 2018 | Markus Borg



Projektplanering och mjukvara: Why care?

1. Programvara utvecklas (nästan) alltid i projektform
2. Programvaruutvecklingsprojekt har frekvent misslyckats sedan 60-talet
 - Kartläggning från Standish Group (2003)
 - 13 522 programvaruprojekt
 - 82% försenade
 - 43% sprängde budget
 - Huvudförklaring inte tekniska problem
 - Mänskliga faktorer dominerar!



LUNDS
UNIVERSITET



Projekt - Ingen allennarådande definition...

”a planned piece of work that has a specific purpose”



Svenska
Akademiens
ordlista

“planerat arbete av större omfattning”

Återkommande egenskaper på projektarbete

- görs ej på rutin - osäkerhet råder
- planering är nödvändigt - även för det osäkra
- finns ett uttalat mål
- förutbestämd tidsram
- resurserna är begränsade



LUNDS
UNIVERSITET

Programvaruprojekt vs. traditionella ingenjörprojekt

Programvara är ingen fysisk produkt – bara information!

- ”osynlig” produkt, framsteg mindre tydliga
- påverkas inte av välkända fysiska lagar
- kan förändras sent – både styrka och utmaning
- komplexitet per \$ hög



**Vi bygger inte ett hus till eller
ännu en bro... Vi utför
innovation på beställning!**



LUNDS
UNIVERSITET

Grundläggande begrepp

Milstolpe = en utvecklingsaktivitets slutpunkt, t.ex.

- Scope freeze
- Code complete
- Conclusion of test

Leverabel = konkret projektresultat som tas emot av någon intressent. Produceras ofta i samband med milstolpe. Exempel:

- Prototyp
- Kravspecifikation 1.0
- Testrapport



LUNDS
UNIVERSITET

		måndag						tisdag						onsdag						torsdag						fredag						lö	sö		
V	tid	8	10	12	13	15	K	8	10	12	13	15	K	8	10	12	13	15	K	8	10	12	13	15	K	8	10	12	13	15	K				
12	Aktivitet	19/3	F1 MA1																																
	Grupp 1-4																																		
	Grupp 5-8																																		
	Grupp 9-12																																		
	Grupp 13-16																																		
13	Aktivitet	26/3	F2 MA1																																
	Grupp 1-4																																		
	Grupp 5-8																																		
	Grupp 9-12																																		
	Grupp 13-16																																		
14	Aktivitet	Annandag påsk						Tentamensperiod						Tentamensperiod						Tentamensperiod						Tentamensperiod									
15	Aktivitet	Tentamensperiod						Tentamensperiod						Tentamensperiod						Tentamensperiod						Tentamensperiod									
16	Aktivitet	16/4	F3 MA1																																
	Grupp 1-4																																		
	Grupp 5-8																																		
	Grupp 9-12																																		
	Grupp 13-16																																		
17	Aktivitet	23/4	F4 MA1																																
	Grupp 1-4																																		
	Grupp 5-8																																		
	Grupp 9-12																																		
	Grupp 13-16																																		
18	Aktivitet	Siste april						1 maj						Kod-jour																					
19	Aktivitet	7/5	F5 MA1																																
20	Aktivitet	14/5	F6																																
21	Aktivitet	21/5	F7 V:A																																
22	Aktivitet	Hemtenta																																	
23	Återkoppling																																		
24	Återkoppling																																		
25	Återkoppling																																		



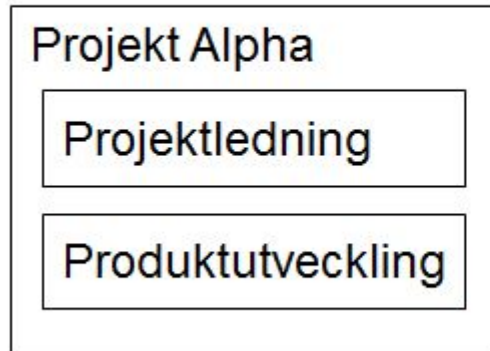
LUNDS
UNIVERSITET

Projektplanering och Projektplaner

ETSA02 Programvaruutveckling – Metodik 2018 | Markus Borg



Vad kommer först: Projektplan eller kravspecifikation?



- Beroende av varandra
- Kraven är en del av produkten. Sista versionen måste sparas.
- Planen hör till organisationen. Erfarenheterna bör sparas.



**Bra planering garanterar
inte lyckade projekt...**

**... men dålig planering
leder ofta till misslyckande!**



LUNDS
UNIVERSITET

Planeringen färdig först vid projektslut

“In preparing for battle I have always found that plans are useless, but planning is indispensable.”

- *Dwight D. Eisenhower*



- **Planering är en iterativ process som pågår under hela projektet**
- **Uppföljning under utvecklingen kritiskt!**



LUNDS
UNIVERSITET

Underskatta inte kommunikation!

Effektiv kommunikation nödvändigt för lyckade projekt

- Fysiska möten bäst

Global software engineering svårt, men allt vanligare

- Videokonferens, telefonmöten, mail, intranät etc.

Outsourcing –
Köpa utvecklingsarbete av
annat bolag
Offshoring –
Utlandsentreprenad, etablera
utvecklingscenter i annat land

“An Empirical Study of Speed and Communication in Globally Distributed Software Development”, Herbsleb and Mockus, 2003

http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1205177&filter%3DAND%28p_IS_Number%3A27132%29

“Conflict Management in Student Groups - A Teacher’s Perspective in Higher Education”, Borg et al., 2011

<http://journals.lub.lu.se/ojs/index.php/hus/article/view/4923>

Förmedla förväntningar och framsteg

- Programvara är en osynlig produkt
 - Utvecklingsarbetet måste aktivt synliggöras
- Förväntningar och framsteg måste kommuniceras
- Bryt ned krav till konkreta arbetspaket
 - Följ upp hur arbetet fortskrider
 - Rapportera kontinuerligt till alla inblandade



LUNDS
UNIVERSITET

KPI - Key Performance Indicator

- Nyckeltal för att värdera en organisations verksamhet
 - Antal nya per vecka: krav, rader kod, testfall
 - Förseningar
 - Försäljningsresultat
 - Fel som hittas under test
 - Rapporterade buggar från kunder



Fördelar

- Enkelt att mäta och förstå
- Kan jämföras över tid och mellan projekt

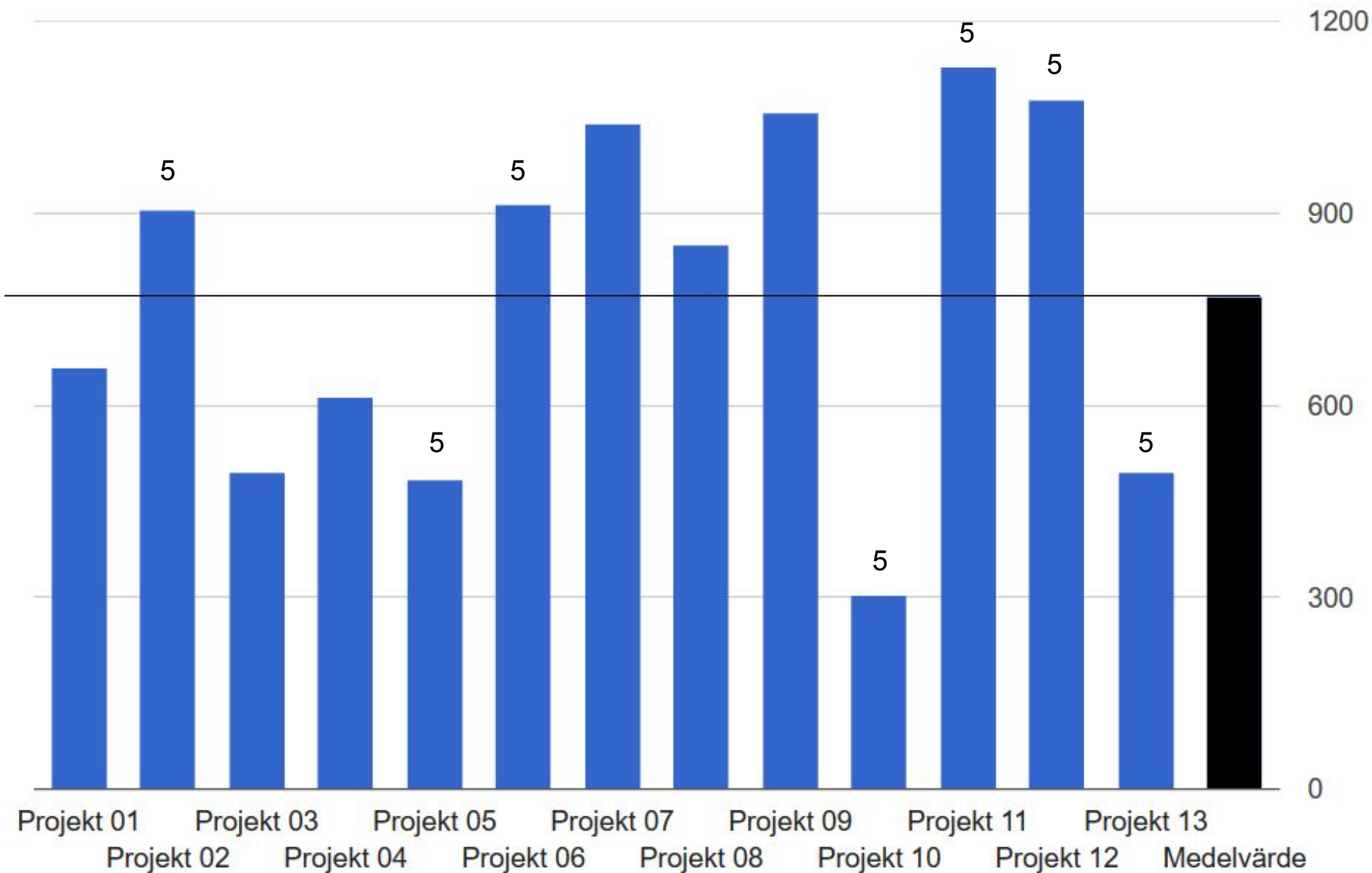
Nackdelar

- Förenklad bild av verkligheten
- Risk för suboptimering



LUNDS
UNIVERSITET

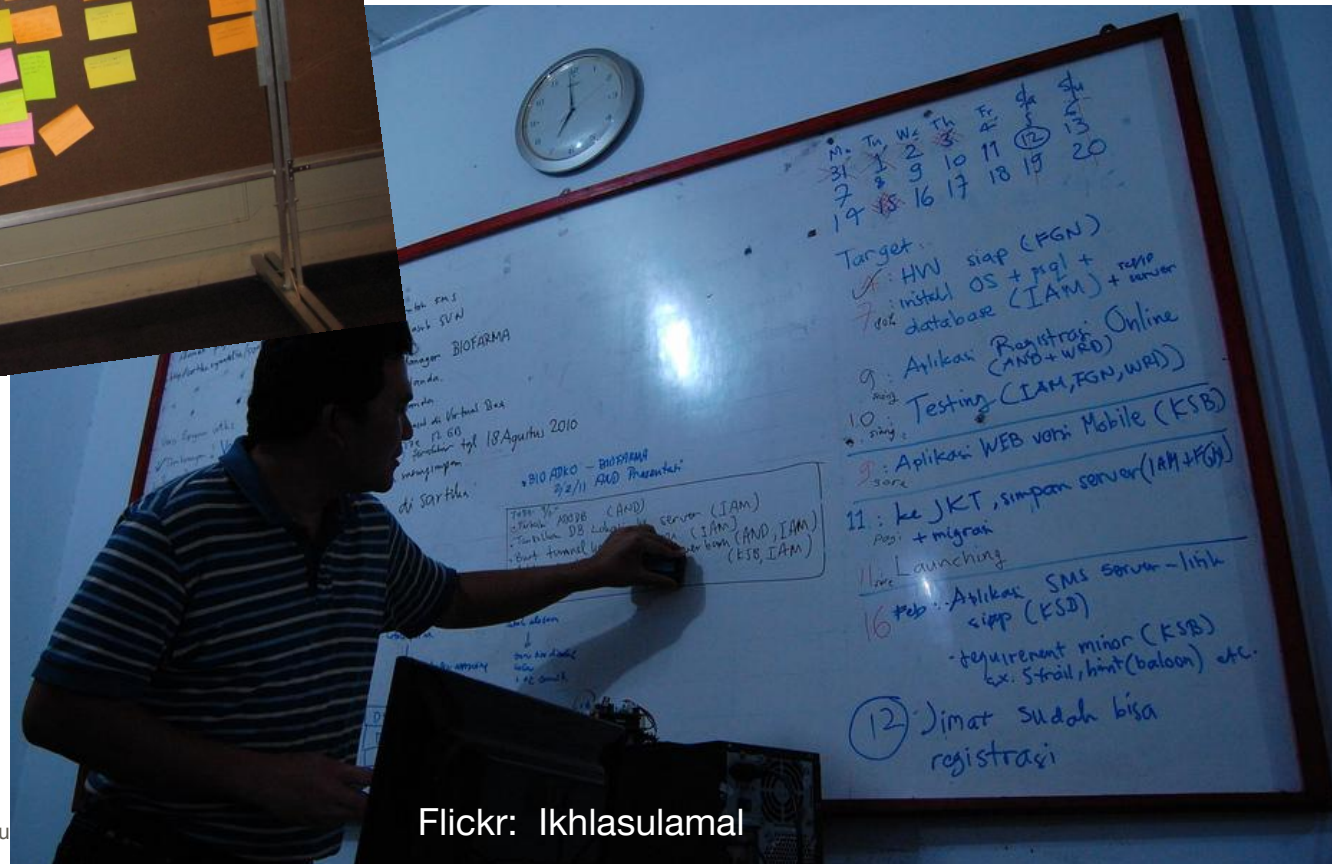
KPI: Kravvolym - Antal ord i tidig version av SRS



Kraftfulla verktyg: Whiteboards och post-its!



Flickr: geodog



Flickr: Ikhlusalamal

Fyra viktiga moment i projektplanering

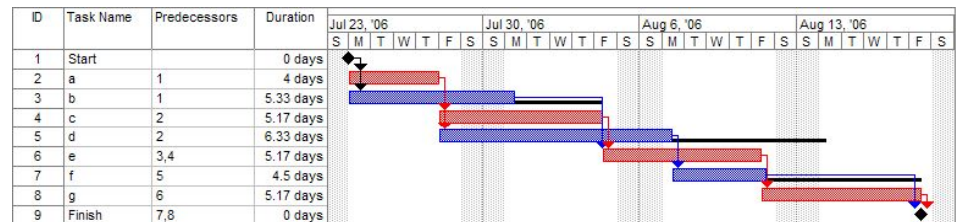
Intressentanalys



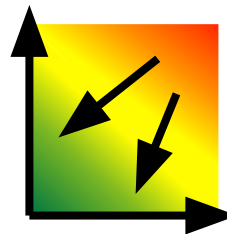
Kostnadsskattning



Schemaläggning



Riskhantering



LUNDS
UNIVERSITET

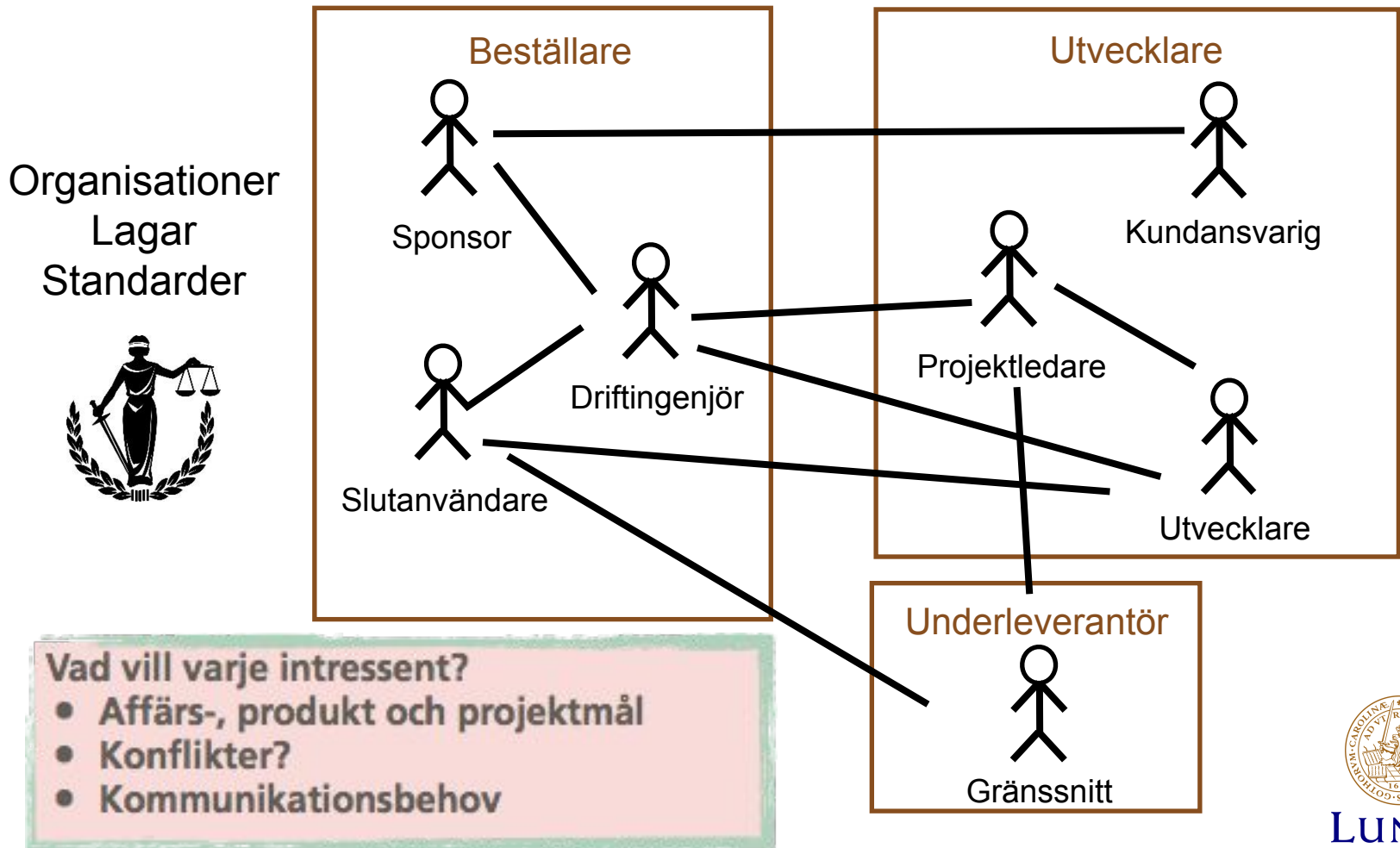
Olika typer av mål

- Affärsmål - i linje med beställarens affärsidé
 - Hur genererar detta intäkter?
 - Varför ser aktieägarna värde i detta?
- Projekt mål - verksamhetens målsättningar
 - Lära sig ny domän, introducera nyanställda, utveckla nya arbetsmetoder, prova ny teknik
- Produktmål - ambitioner med den nya produkten
 - Vad utmärker produkten på marknaden?
 - Vilken nytta erbjuds användarna?

Uppdelningen gör att motsättningar kan identifieras!



Intressentanalys (stakeholder analysis)



Kostnadsskattning

Vid programvaruutveckling
domineras kostnaderna av persontid

- Kostnadsskattning => tidsuppskattning



Enkelt mått på programvarans storlek

- rader källkod (lines of code)

Naivt mått på utvecklarens produktivitet

- rader källkod per personmånad

Varierar enormt!

Stora komplexa system:

~30 rader/personmånad

Enkel välkänd domän:

~900 rader/personmånad

Programmerarens
förmåga kan påverka
med en faktor 10



LUNDS
UNIVERSITET

Kostnadsskattning – Tre metoder

Expertbedömning

- Flera erfarna personer gör kvalificerade gissningar
- Top-down: övergripande funktioner → subfunktioner → integration
- Bottom-up: komponenter → subsystem → system

Estimat baserade på analogier

- Jämför med tidigare utvecklingsprojekt
- Liknande storlek? Komplexitet? Motsvarande utvecklarkompetens?

Algebraiska metoder

- Räkna ut ett estimat, till exempel: $\text{kostnad} = A \times \text{Size}^B \times M$
A = komplexitet, B = extrakostnad för stora system, M = mognad
- Parametrarna bestäms baserat på databas med historiska projekt

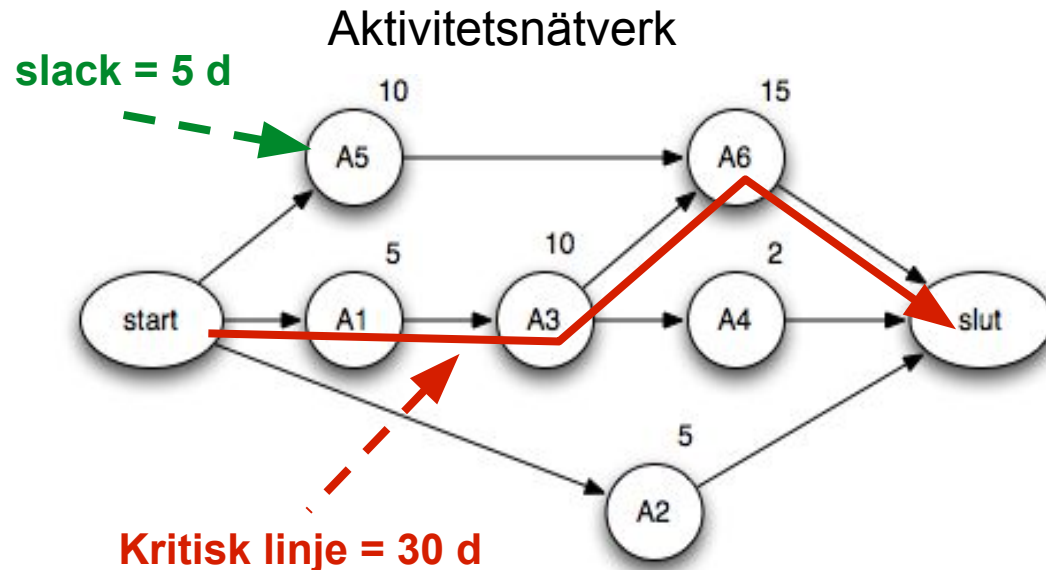


LUNDS
UNIVERSITET

Schemaläggning - Aktivitetsnätverk

- Bryt ned projekt i arbetspaket
- Estimera tidsåtgång och beroenden
- Identifiera kritisk ledtid, dvs. minimal genomförandetid

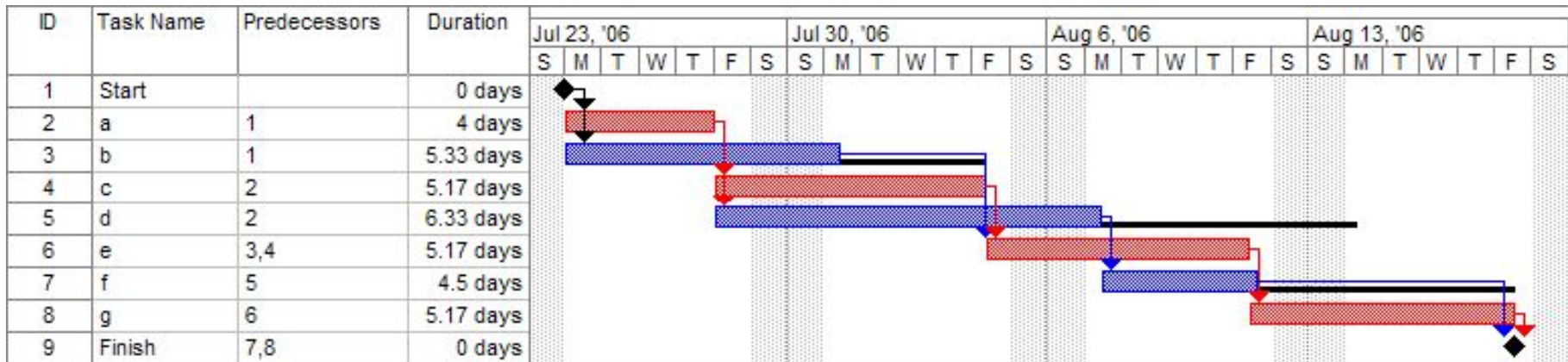
Aktivitet	Tid (d)	Beroenden
A1	5	
A2	5	
A3	10	A1
A4	2	A3
A5	10	
A6	15	A3, A5



- Störningar på **kritisk linje** försenar projektet
- Övriga aktiviteter har **slack**, dvs. utrymme för försening

Schemaläggning – Gantt-diagram

- Horisontella stapeldiagram med tidsaxel
- En managementrevolution vid 1900-talets början!



- Hämtad från Wikipedia, skapad med Microsoft Project
- Kritisk väg presenteras i rött
- Slack representeras av svart linje



LUNDS
UNIVERSITET

”Naturlagar” inom software engineering

Mest på skoj, men sätter fingret på upplevda fenomen.
Urval, fritt översatta:

Parkinsons lag

”En arbetsuppgift kommer att ta den tid som är avsatt för ändamålet.”

Hofstadters lag

“En arbetsuppgift tar alltid längre tid än du förväntar dig, även om du tar Hofstadters lag med i beräkningen.”

Brookes lag

”Att tilldela fler utvecklare till ett försenat projekt försenar det ytterligare.”

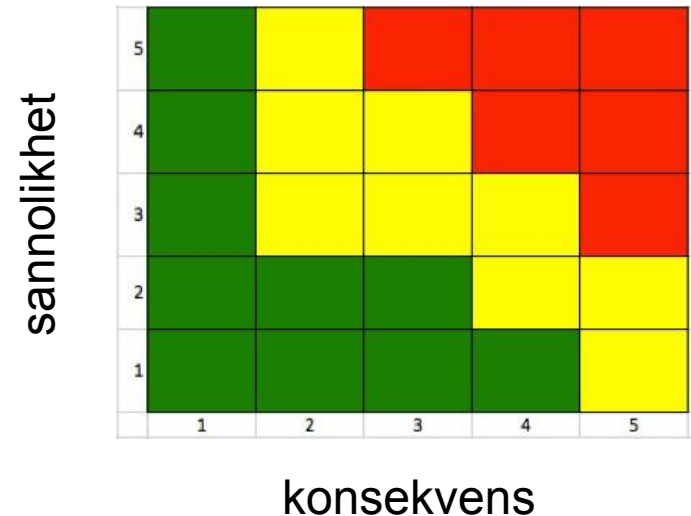
Riskhantering

Definition av risk

sannolikhet för
oönskad konsekvens

×

konsekvensens storlek



Olika risktyper har olika påverkan

- **Projektrisker:** projektplan, tillgängliga resurser
- **Produktrisker:** programvaran som utvecklas
- **Affärsrisker:** påverkar utvecklingsorganisationen



LUNDS
UNIVERSITET

Riskhanteringsprocessen

Aktivitet

1) Identifiera
risker

2) Bedöm
risker

3) Behandla
risker

4) Följ upp risker
under projektet

Leverabel

Lista med
risker

Priorite-
rade
risker

Planer för
risker

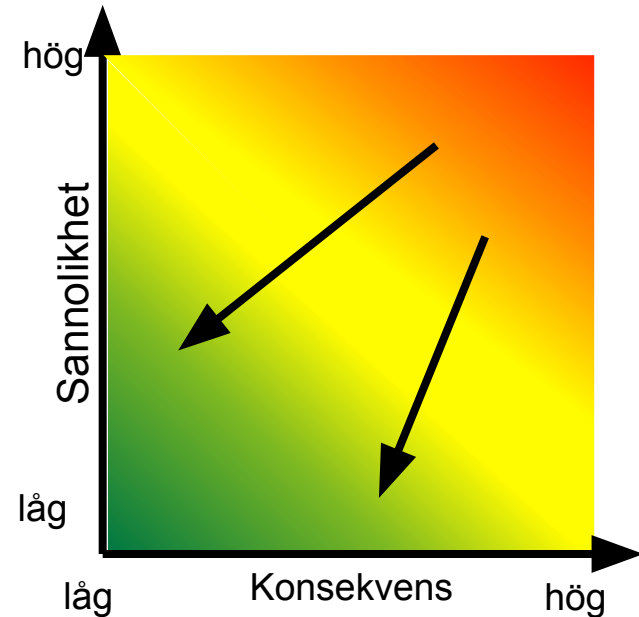


LUNDS
UNIVERSITET

Riskhantering i praktiken

Strategier

- Reducera konsekvens
- Minska sannolikhet
- Alternativ (plan B)



Riskkälla	S	K	Risk (S x K)	Strategi
Hårdvara försenad	2	5	10	Undersöka alternativ
				Konstruera simulator
Sjukskrivningar	1	2	2	Begränsa övertid
Krav förändras	4	3	12	Veckomöten med kund

Innehåll i en projektplan

Inledning

projektmodell, övergripande produktbeskrivning, målsättningar, begränsningar

Projektorganisation

utvecklingsorganisation, testorganisation, andra intressenter

Hårdvara och programvara

Resurser som krävs för projektets genomförande

Arbetsnedbrytning

aktiviteter, leverabler, milstolpar

Tidplan

när varje aktivitet påbörjas och avslutas, när varje milstolpe ska uppnås

Uppföljning och rapportering

hur framsteg mäts och hur det kommuniceras

Risikanalys



LUNDS
UNIVERSITET

Tidpunkt, kostnad eller kvalitet?

Tidpunkt
När ska vi leverera?



Kostnad
Vad får det
få kosta?

Kvalitet
Hur bra ska
det bli?

Tre önskvärda
egenskaper:

- Leverans i tid
- Utveckling inom budget
- Programvara med god kvalitet

Bara en eller två kan
prioriteras!



LUNDS
UNIVERSITET

Sammanfattning projektplanering

- Programvaruprojekt speciella eftersom de innebär komplex innovation av osynlig produkt
- Planering pågår till projektet är avslutat
- Fyra centrala aktiviteter i projektplanering: intressentanalys, kostnadsskattning, schemaläggning och riskhantering
- Projektplanen beskriver bl.a. projektorganisation, arbetsnedbrytning, tidplan och riskanalys



LUNDS
UNIVERSITET

Statisk testning - Granskning

ETSA02 Programvaruutveckling – Metodik 2018 | Markus Borg

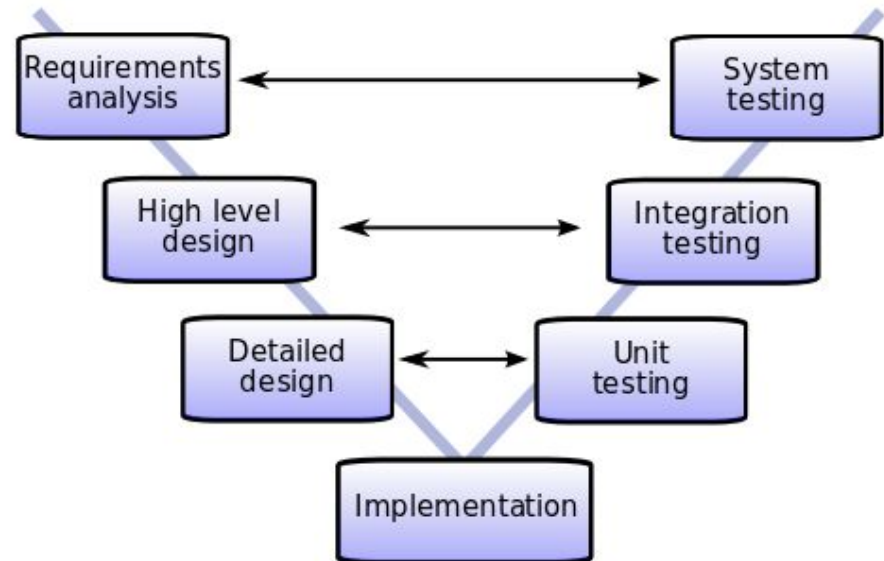


Granskningar – grundläggande idé

Hitta fel tidigt utan att exekvera kod – dvs. statisk testning

Alla artefakter kan granskas (kravspecifikation, testspecifikation, design, källkod, testfall etc.)

- Läs artefakt på ett strukturerat sätt
- Rätt personer ska läsa
- Personerna ska läsa på rätt sätt
- Alla viktiga delar av dokumenten ska läsas



Typer av granskningar

- Kodgranskning
 - Systematisk undersökning av källkod.
- Inspektion
 - Formell insats för att identifiera brister eller säkerställa kvalitet enligt väldefinierad process.
- Audit
 - Genomlysning av oberoende organisation för att säkerställa att produkt eller process uppfyller standarder, regelverk, lagstiftning, etc.



LUNDS
UNIVERSITET

Kodgranskning - Vad och varför?

- Okulärbesiktning av någon annans källkod som komplement till dynamisk testning
- Huvudsyften
 - Identifiera fel
 - Upprätthålla konventioner
 - Identifiera sårbarheter
 - Sprid kunskap om koden



LUNDS
UNIVERSITET

Kodgranskning - När och hur?

- Utvecklingsprocessen kan specificera kodgranskning vid olika tillfällen på olika vis. Till exempel:
- Innan commit and push (eller pull request)
 - Annan utvecklare måste granska innan kod hamnar i kod-repository
- Efter commit and push
 - Annan utvecklare måste granska ny kod som integrerats i kod-repository
- Inför release
 - Flera utvecklare genomför formell inspektion



LUNDS
UNIVERSITET

Kodgranskning - Tips!

- Kodgranskning är bra och viktigt. Källkoden blir bättre och kunskap sprids i organisationen.
- Mängder av “best practices” på Internet. Exempelvis:
 - Granska inte för många rader kod åt gången (<400)
 - Stressa inte. Det ska inte gå fort.
 - Granska inte för längre åt gången (<1 h)
 - Notera brister i en granskningslogg
 - Planera för hur identifierade brister ska åtgärdas



LUNDS
UNIVERSITET

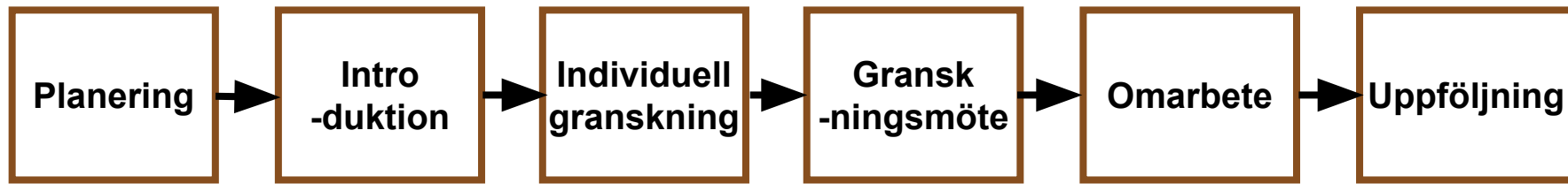
Kodgranskning - Verktøgsstöd

- Vissa typer av brister kan man verktyg för att identifiera
 - Låt människorna fokusera på resten
- Statisk kodanalys (Lab 4)
 - Verktøg som analyserar källkoden utan att exekvera den (antingen källkod eller kompilerad kod)
 - Kan identifiera när kod bryter mot konventioner
 - Kan identifiera kodkonstruktioner som ofta leder till buggar
 - Kan identifiera sårbarheter



LUNDS
UNIVERSITET

Inspektioner (för både dokument och kod)

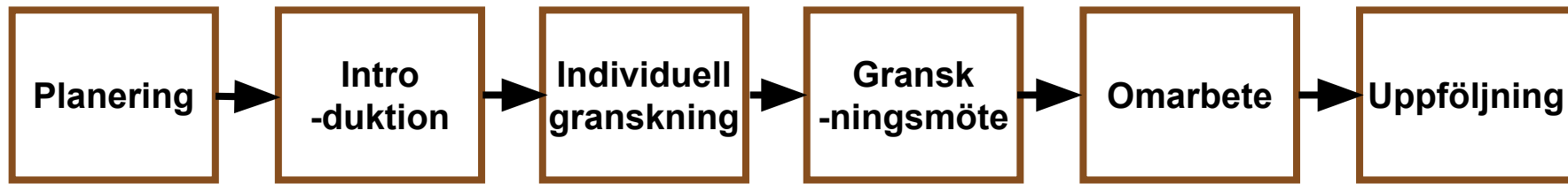


- Formell process för strikt granskning
- Utvecklad på 70-talet av IBM
- Ofta processkrav i kritiska domäner
- Resultat dokumenteras noggrant
- Övning 4
 - Granskningsprotokollet del i Beta Release

The screenshot shows a web-based form titled "Granskningsprotokoll för Dokument 0.XX". It includes a menu bar (File, Edit, View, Insert, Format, Data, Tools, Add-ons, Help) and a toolbar with various icons. The form is divided into several sections:

- Grupp**: A table with columns A through F.
- Dokument**: A section for document details, including "Vilket dokument och dokumentversion som har granskats - Länka om på webben", "Datum", "Granskare", and "Dokumenterat av".
- Uppföljning**: A section for follow-up, including "Person eller grupp som följt upp åtgärder, samt datum för uppföljning".
- Prioriteter**: A table with columns A, B, and C, and a description of priorities.
- Checklista**: A section for a checklist, including "Namn och länk till eller referens till checklista som använts vid granskningen".
- Problem**: A table with columns for Problem #, Position i dokument, Problemtyp, Problem, Prioritet, and Uppföljning.

Inspektioner - Steg 1-3

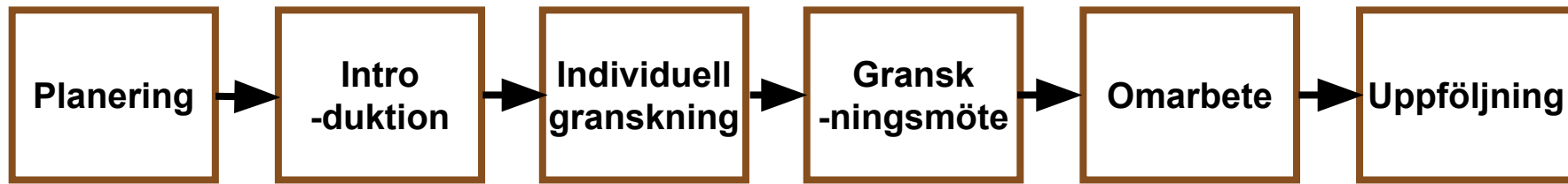


- Planering
 - Förbered material, bjud in personer till möte
- Introduktion
 - Instruera granskarna, tilldela granskningsroller
- Individuell granskning
 - Granskarna identifierar brister
 - Bristerna dokumenteras noga



LUNDS
UNIVERSITET

Inspektioner - Steg 4-6



- Granskningsmöte

- Alla samlas på möte. Artefakten går igenom från början till slut.
- Defekter funna vid individuell granskning dokumenteras.

- Omarbete

- Ansvarig utvecklare åtgärdar bristerna

- Uppföljning



LUNDS
UNIVERSITET

Roller vid granskningmötet

- Moderator
 - Leder mötet.
 - Styr genomläsning av artefakt från början till slut.
- Sekreterare
 - Författar granskningsprotokollet
- Författare (skapare av artefakten)
 - Svarar på eventuella frågor som dyker upp
- Granskare
 - Personer som genomfört individuell granskning



LUNDS
UNIVERSITET

Lästekniker vid individuell granskning

Ad-hoc

- Upp till granskaren

Checklist-baserad

- Stöd av en checklista
- Vanligen framtagen av kvalitetsingenjörer

Scenario-baserad

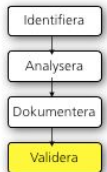
- Följ ett användningsscenario under granskningen

Perspektiv-baserad

- Granska som en specifik roll: användare, testare, operatör, utvecklare, etc.

Checklista för granskning för kursens projekt

1. Saknas några krav?
2. Är samtliga krav nödvändiga?
3. Finns det några motstridiga krav?
4. Kan samtliga krav verifieras?
5. Är samtliga krav tydligt formulerade eller kan några krav misstolkas?
6. Finns samtliga nödvändiga definitioner?
7. Är det möjligt för dokumentets målgrupp att förstå dokumentet?
8. Följer kravspecifikationen sin dokumentmall?
9. Är något krav formulerat för detaljerat?
10. Har något krav formulerats på för hög abstraktionsnivå?
11. Är all text och illustrationer nödvändiga?
12. Har samtliga krav unika identifierare?

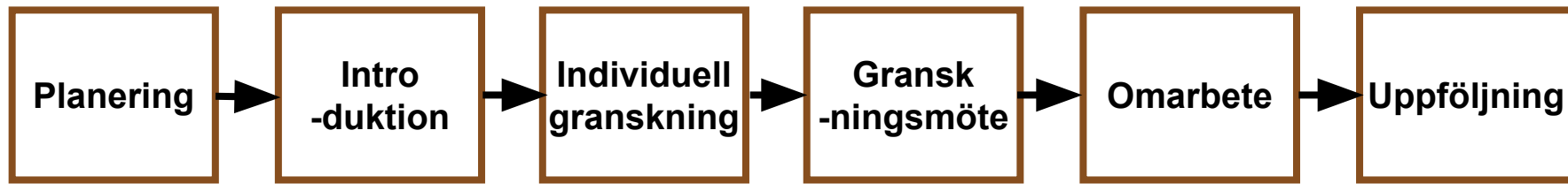


Guidat alternativ



LUNDS
UNIVERSITET

Vad kostar det?



- Planering och introduktion: ? h
- Individuell granskning:
 - Kravspecifikation: 5 sid/h
 - Design: 4 sid/h
 - Källkod: 150 rader/h (utan kommentarer)
 - Testdokumentation: 4 sid/h

**Bjud in till gransknings-
möte först när det är
meningsfullt!**

(Ebenau et.al., Software
Inspection Process,
1994)

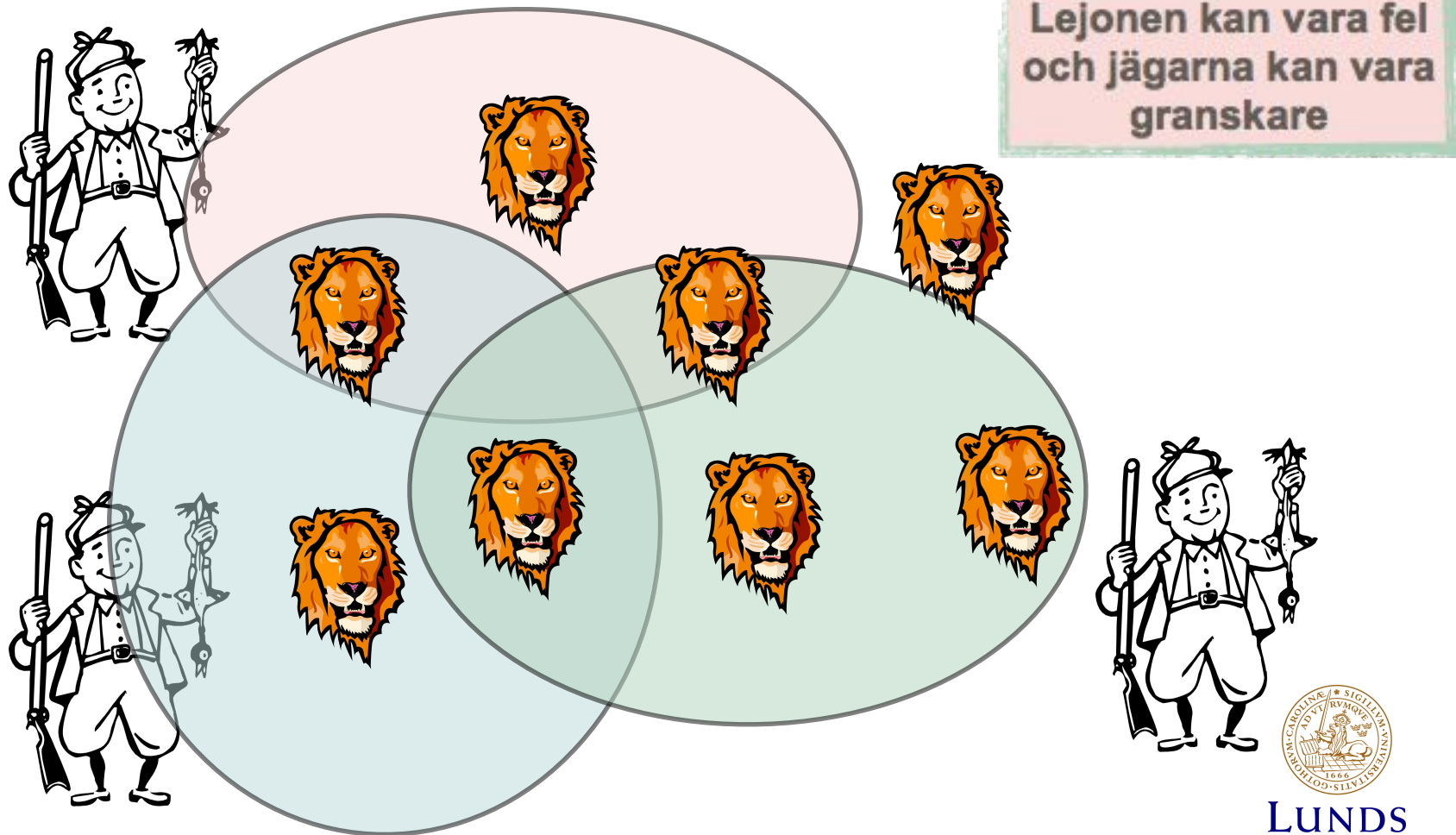
- Granskningsmöte: 4-10 personer \times 2 h
- Omarbete och uppföljning: ? h



LUNDS
UNIVERSITET

Har vi hittat alla brister

- eller hur många lejon finns det i skogen?



Capture-recapture

En metod från ekologin för att estimerar
djurpopulation

Antag två likvärdiga granskare samt brister som är
lika enkla att finna, samt:

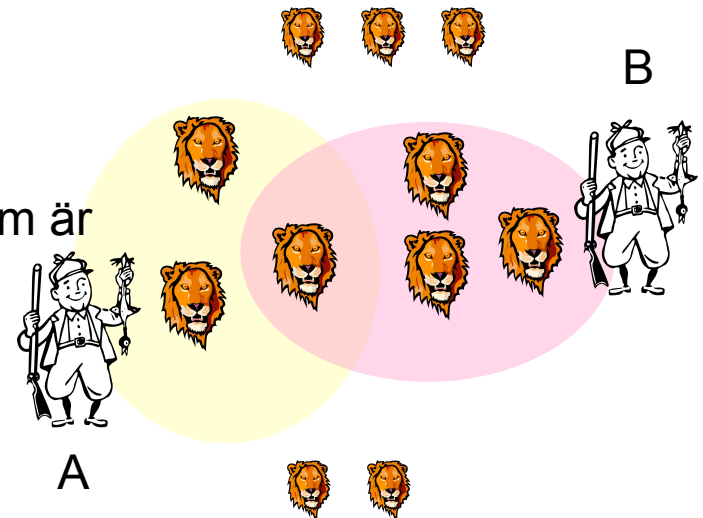
N = totalt antal brister

N_A = antal brister som granskare A hittar

N_B = antal brister som granskare B hittar

N_{AB} = antal brister som båda hittar

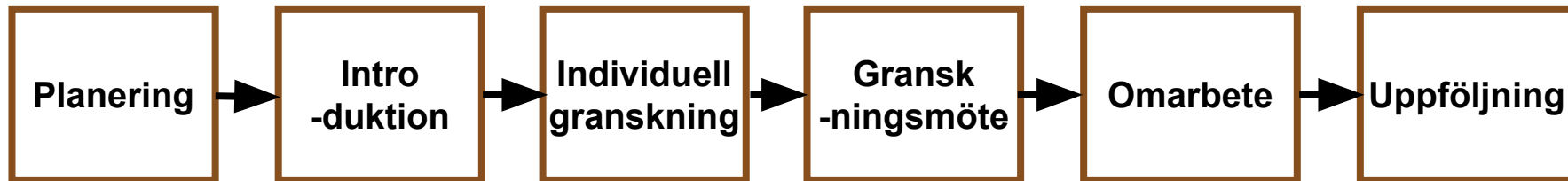
Andel brister som båda hittar bland granskare As
resultat (N_{AB} / N_A) motsvarar andelen brister
granskare B hittade bland samtliga (N_B / N)



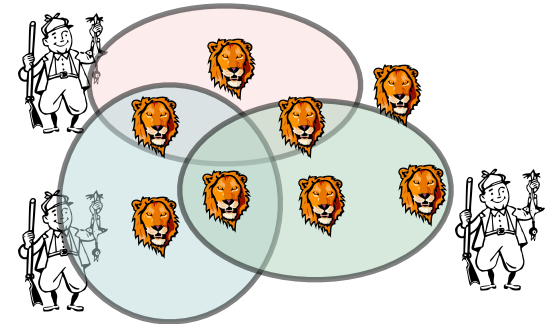
$$\frac{N_{AB}}{N_A} = \frac{N_B}{N}$$

$$N = \frac{N_A \times N_B}{N_{AB}} = \frac{3 \times 4}{1} = 12$$

Statisk testning - sammanfattning



- Systematisk metod för att identifiera brister i artefakter utan exekvering
 - Kodgranskning
 - Inspektion
 - Audit
- Använd verktyg för statisk kodanalys
- Granskningsprocesser är en naturlig del av kvalitetssäkrande arbete
 - Dyrt men bra!



LUNDS
UNIVERSITET



LUNDS
UNIVERSITET

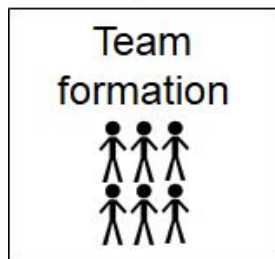
Robotprojekten

Programvaruutveckling - Metodik 2018 | Markus Borg

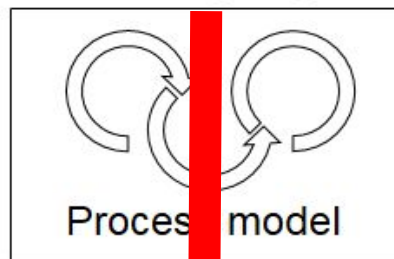


		måndag						tisdag						onsdag						torsdag						fredag						lö	sö			
V	tid	8	10	12	13	15	K	8	10	12	13	15	K	8	10	12	13	15	K	8	10	12	13	15	K	8	10	12	13	15	K					
12	Aktivitet	19/3	F1 MA1														Lab1			Ö1						L1										
	Grupp 1-4																Alfa	PW		3308																
	Grupp 5-8																Falk	PW		2116																
	Grupp 9-12																	Alfa		PW	3308															
	Grupp 13-16																	Beta		PW	2116															
13	Aktivitet	26/3	F2 MA1														Lab2			Ö2						Lång fredag										
	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										
15	Aktivitet	Tentamensperiod						Tentamensperiod						Tentamensperiod						Tentamensperiod						L2	Tentamensperiod									
16	Aktivitet	16/4	F3 MA1														Lab3			Ö3						L3										
	Grupp 1-4																Alfa			3308																
	Grupp 5-8																Falk			2116																
	Grupp 9-12																	Alfa			3308															
	Grupp 13-16																	Gamma			2116															
17	Aktivitet	23/4	F4 MA1														Lab4			Ö4						L5										
	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																						
19	Aktivitet	7/5	F5 MA1														Kristi himmelsfärdsdag												L6							
20	Aktivitet	14/5	F6 MA4																						L8	Lundakarneval										
21	Aktivitet	21/5	F7 V:A																																	
22	Aktivitet													Hemtenta																						
23	Återkoppling																																			
24	Återkoppling																																			
25	Återkoppling																			Midsommarafton																

Project
inception



Engineering, monetizing,
strategizing



Post release

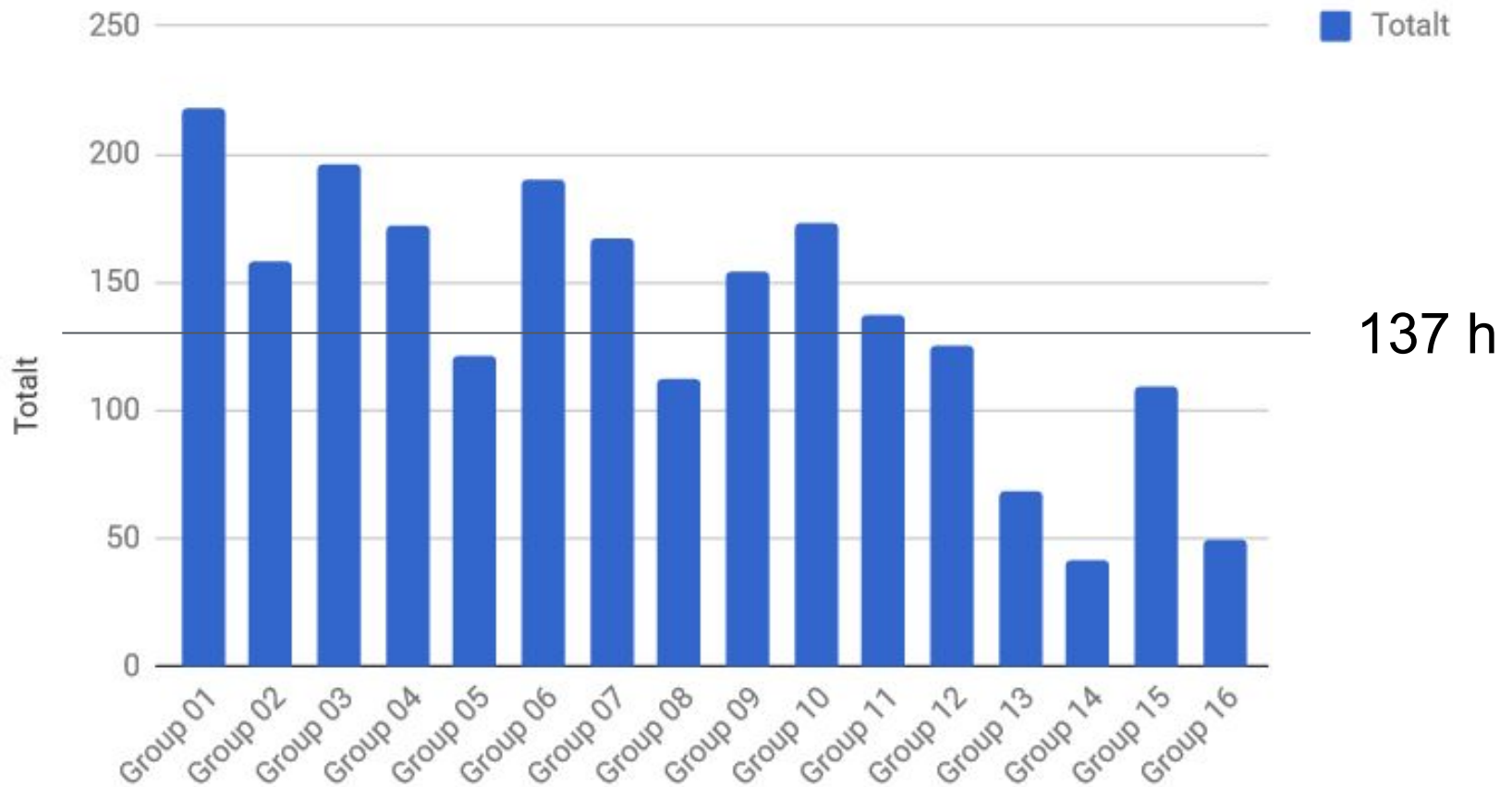


LU Rumble

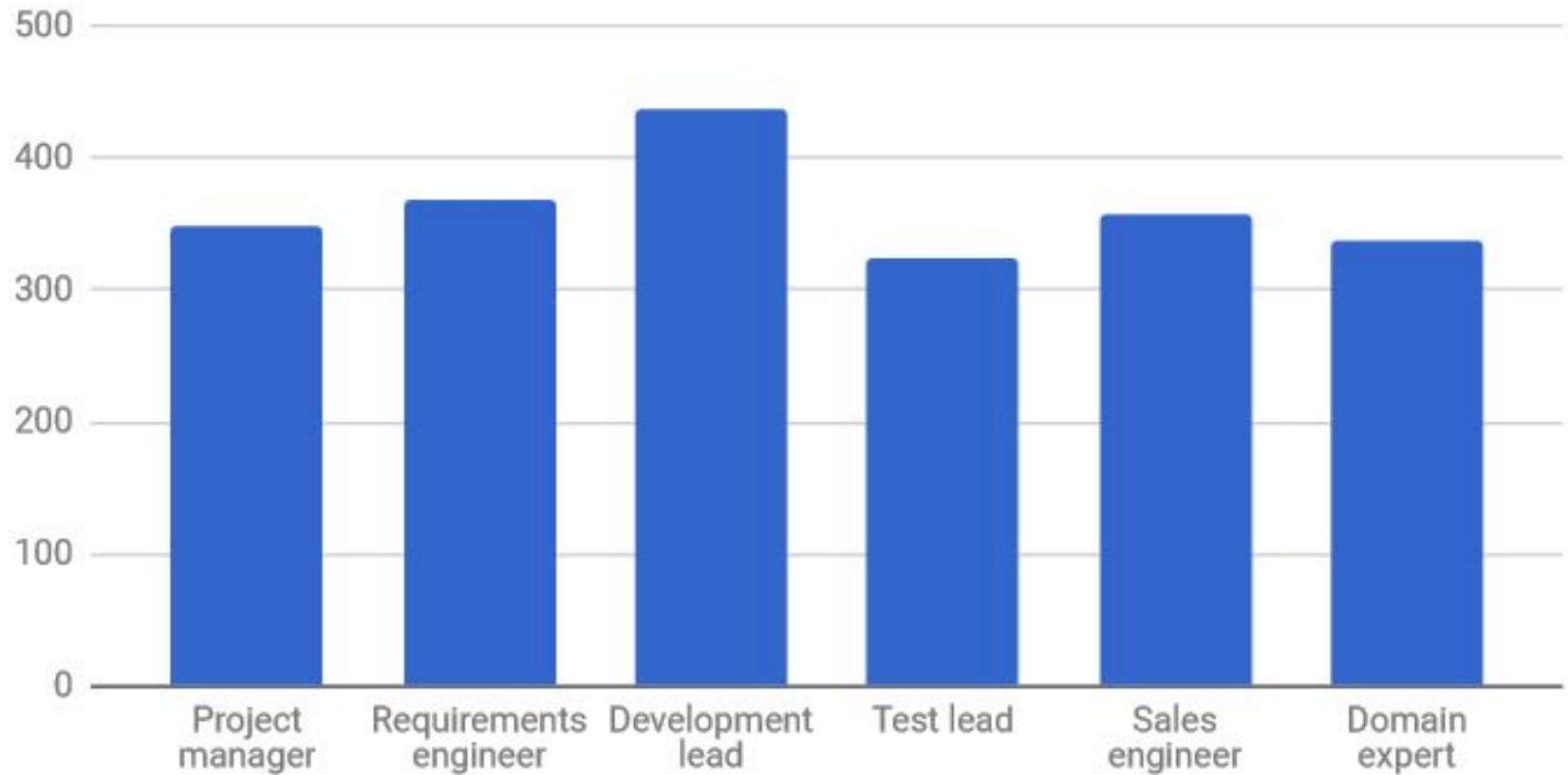


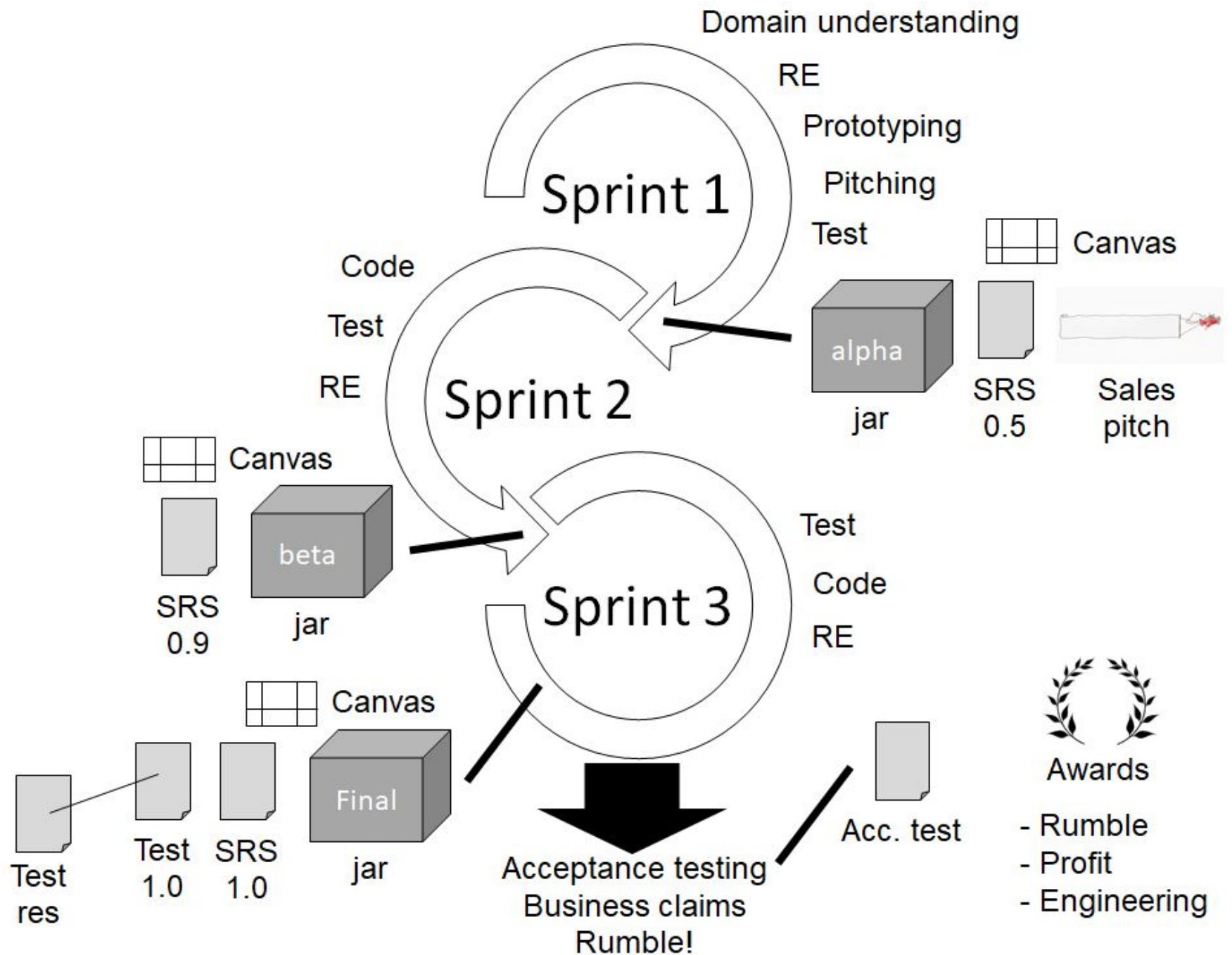
Tidrapporter - Total tid per grupp

Totalt



Tidrapporter - Total tid per roll



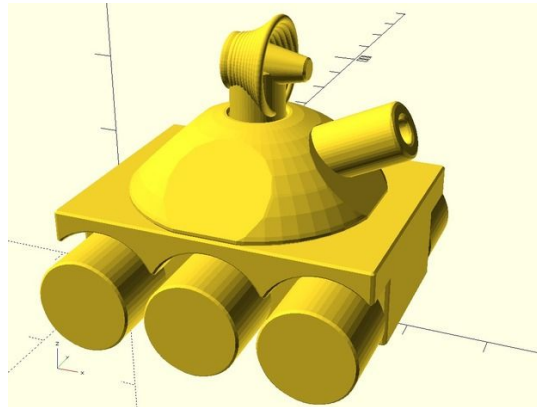


- Verkt yg f r kvalitetss kring (Lab 4)
- Granskningsm te SRS ( 4)

- Aff rsrelation best llare-leverant r

Engineering

Monetizing



Strategizing

- Fundera, fundera, fundera...

Detta  r p  g ng:

På gång gällande ETSA02 RoboTalk

- v1.2 närmar sig
- Räkna med att det införs krav på lagfärger



LUNDS
UNIVERSITET

L5: Fredag kl 23.59

Beta Release - Minimum Viable Product

- **SRS v0.9 (detaljerade kraven specificerade)**
- Granskningsprotokoll från Övning 4
- STS v0.9 (notera ny template på Google Drive)
- Lean canvas (uppdaterad vid behov)
- **Robot som jar-fil**
 - Med källkod (+testkod, klassdiagram, Javadoc)
 - **Utan källkod**

ARTEFAKTER I RÖTT ÄVEN TILL KUND



LUNDS
UNIVERSITET