



LUNDS
UNIVERSITET

ETSA02

Programvaruutveckling – Metodik

Föreläsning 1 – Markus Borg





Flickr: nearsoft



Flickr: canadianveggie

Flickr: carlcollins



Flickr: nearsoft



Flickr: m-i-k-e

Pedagogisk utmaning!

Kan man förstå software engineering utan att ha upplevt stora programvaruprojekt?



Vi genomför ett utvecklingsprojekt och belyser det med teori och reflektioner

Kan man förstå vad som händer i stora programvaruprojekt utan att ha studerat software engineering?



LUNDS
UNIVERSITET

Markus Borg - CV



Civilingenjör Data teknik

2002-2007

- Programvaruutveckling
- Exjobb reglerteknik/datavetenskap



LUNDIENS
UNIVERSITET

Utvecklingsingenjör ABB

2007-2010



- Processautomation

Doktorand Programvarusystem

2010-2015



LUNDIENS
UNIVERSITET

- Maskininlärningstekniker för systemevolution

Forskare, RISE SICS AB

2015-

- Utveckling av maskininlärningsbaserade system

20% Adjungerad lektor, Datavetenskap 2017-



LUNDIENS
UNIVERSITET

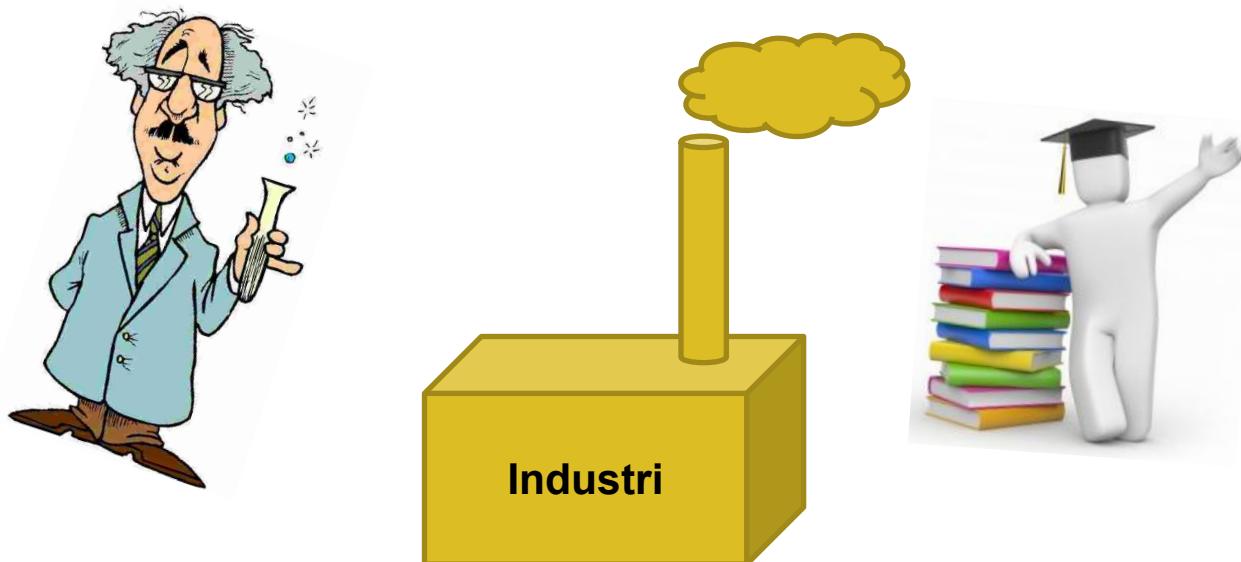
Agenda föreläsning 1

Allmänt om kursen

Projektuppgiften

Kravhantering

I pausen: Bilda projektgrupper



LUNDS
UNIVERSITET



LUNDS
UNIVERSITET

Allmänt om ETSA02



Personal



Markus Borg
kursansvarig,
föreläsningar,
övningar,
projekthandledning



Elizabeth Bjarnason
övningar,
projekthandledning



Anna Axelsson
övningar,
projekthandledning



Sergio Rico
övningar,
projekthandledning

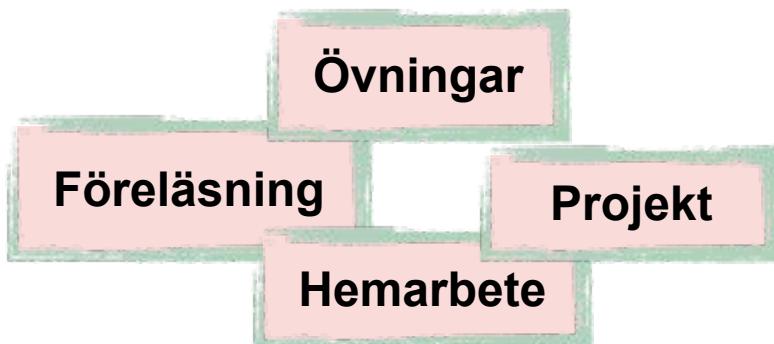


**LUNDS
UNIVERSITET**

Kursen

Innehåll

- Kravhantering
- Projektplanering
- Arkitektur
- Testning
- Processmodeller



Formalia

- 6 hp
- Obligatorisk för C1
Alternativobligatorisk för I3
- Moment
 - 7 Föreläsningar
 - 8 Övningar
 - » 4 i datorsal
 - Projekt
 - Hemtentamen

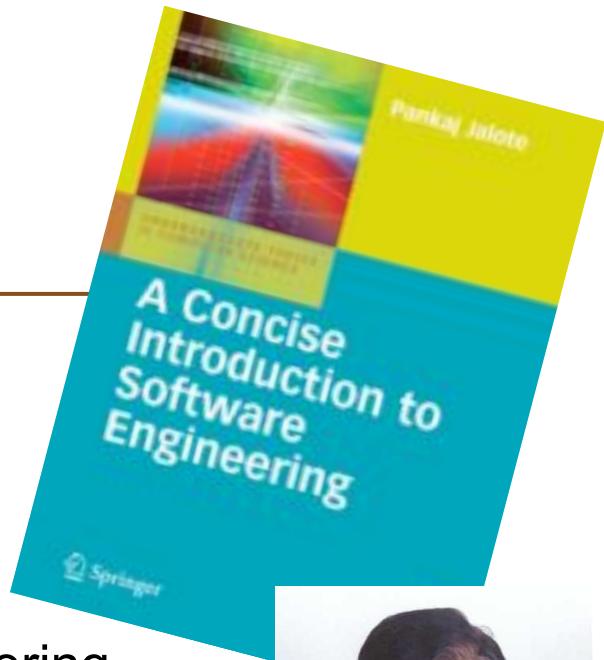


LUNDS
UNIVERSITET

Kurslitteratur

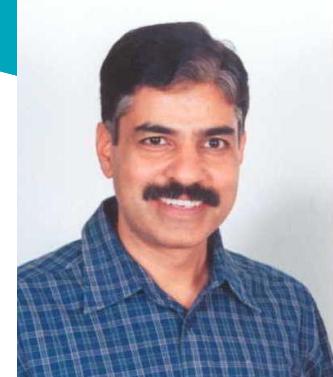
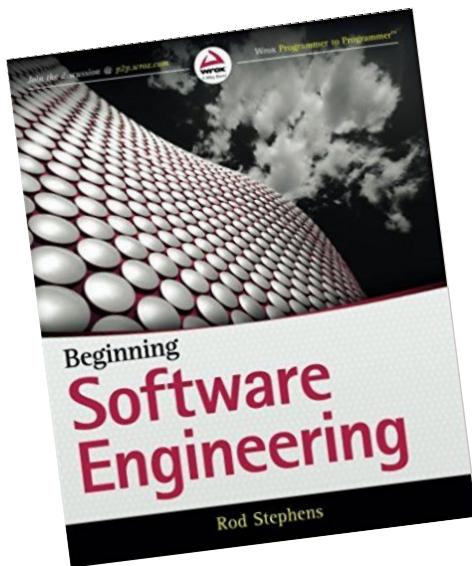
1. **Formell bok**

Pankaj Jalote, *A Concise Introduction to Software Engineering*, Springer, 2008.



2. **Ny bok (formell nästa gång)**

Rod Stephens, *Beginning Software Engineering*, Wrox, 2015.



LUNDS
UNIVERSITET

Examination

- A. Projektarbete – 3 hp
- B. 34 timmar hemtenta - 3 hp

Exempel från tidigare hemtenta

U3 Kravgranskning (10 av 60 p)

Beskriv hur en kravgranskning går till och förklara målsättning, varför denna typ av granskning är viktig, vem som bör delta, samt vilka typer av fel man bör leta efter.

Diskutera vilka svårigheter som finns då man vill införa denna typ av granskningar i en organisation samt ange tänkbara lösningar på dessa problem. Ange också om det finns några alternativ till denna typ av granskning och vilka dessa i så fall är.



**LUNDS
UNIVERSITET**

Hälsningar från olika håll...

Tidigare studenter

- Kursboken kan upplevas som svår - det är den inte
- Projektet innebär att man ofta känner osäkerhet
- Det är viktigt att alla i projektgruppen har koll på tidplanen
- vem, vad, när & varför
- Bred medverkan i projektet ger stor fördel på tentan

Studenter som börjat jobba i stora företag

- Kursmomenten ibland frustrerande, men få kurser lika tydligt kopplade till arbetsrollen som ingenjör

Rekryterare i industrin

- Kurser med projekt som utförs i grupp väldigt viktigt
 - nyutexaminerade studenter idag är mycket bättre förberedda för projektarbete än för 10 år sedan



LUNDS
UNIVERSITET

Om projektuppgiften



I praktiken

I grupper om 6 personer:

Genomför ett litet utvecklingsprojekt från början till slut

- Marknadsanalys
- Kravidentificering
- Design
- Implementation
- Testning
- Leverans
- Tidrapportering
- Prissättning

Leverabler

- Sales pitch
- Kravspecifikation
- Testspecifikation
- Testrapporter
- Källkod
- jar-fil

Tekniska erfarenheter

- Java
- Eclipse
- JUnit
- git
- GitHub
- Maven
- SonarQube



Projektgrupperna ska utveckla robotar för LU Rumble 21 maj

- Ingen grupp tävlar med sin egen robot.
- Varje grupp säljer sin robot till andra grupper
- Varje grupp köper in andra robotar för att bilda ett lag



LUNDS
UNIVERSITET

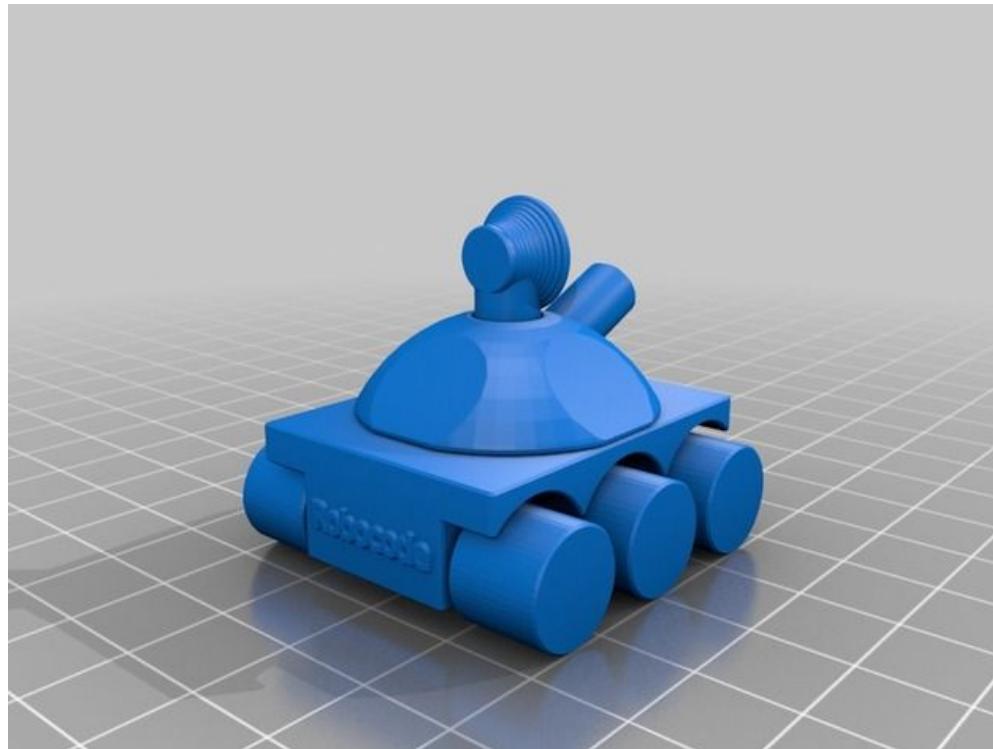
Robocode - Build the best, destroy the rest!

- Spelplattform utvecklat för att träna programmering
 - Event-driven programmering
- Påbörjades 2001, ägdes därefter av IBM, öppen källkod sedan 2005
 - Underhålls aktivt av Flemming N. Larsen
- Enkelt att börja, svårt att sluta
- Ingen övre gräns för hur avancerade robotarna kan bli
 - Maskininlärning
 - Adaptivt beteende
 - Genetisk programmering
 - Flockbeteende



LUND
UNIVERSITET

Produkten



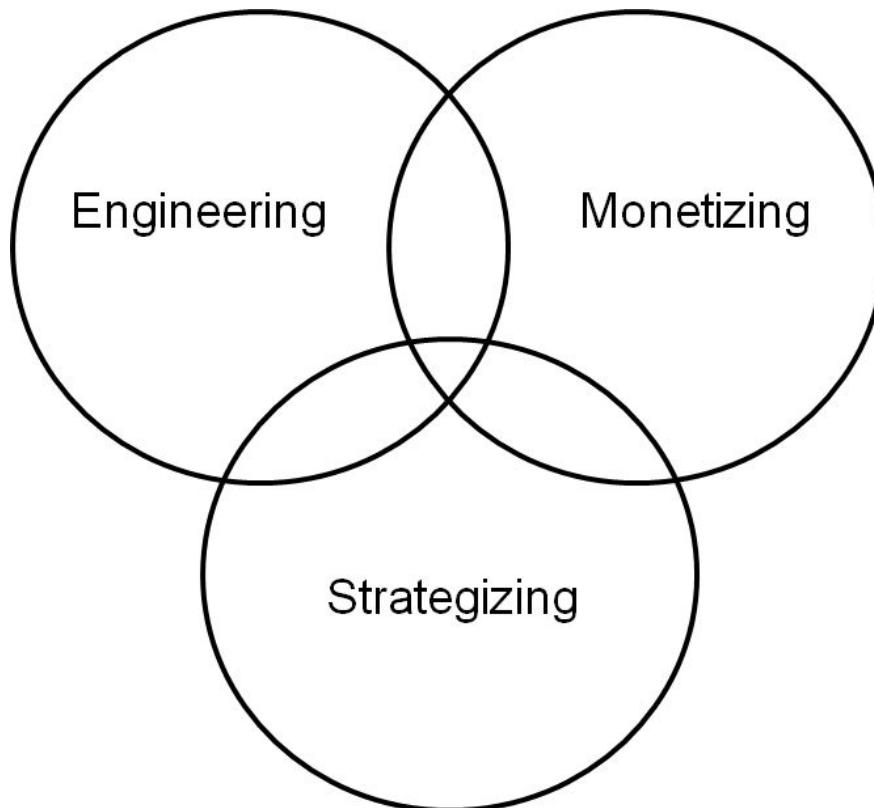
- Sex hjul, en radar, en kanon och lite kommunikation



LUNDS
UNIVERSITET

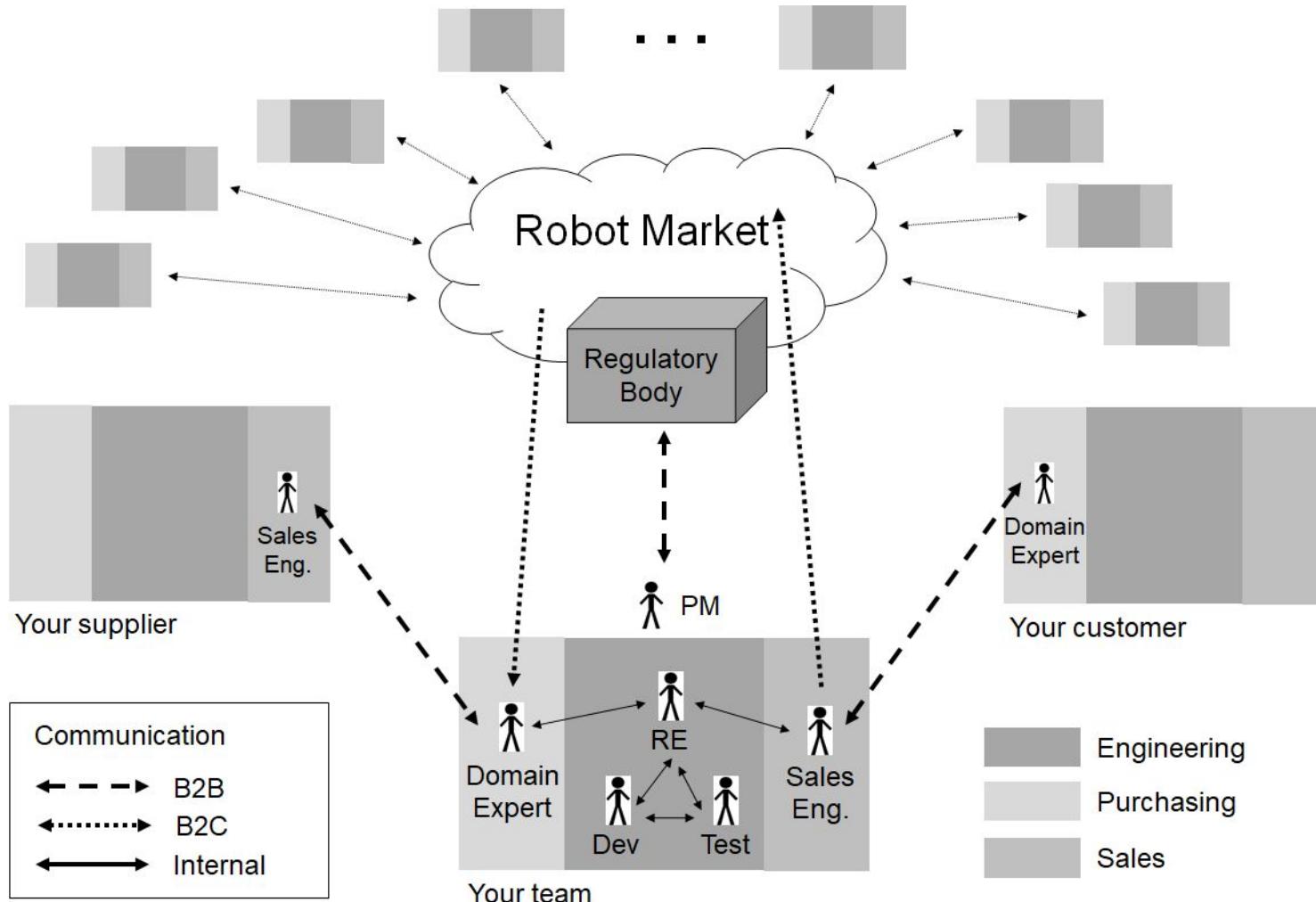


Tre moment i projekten



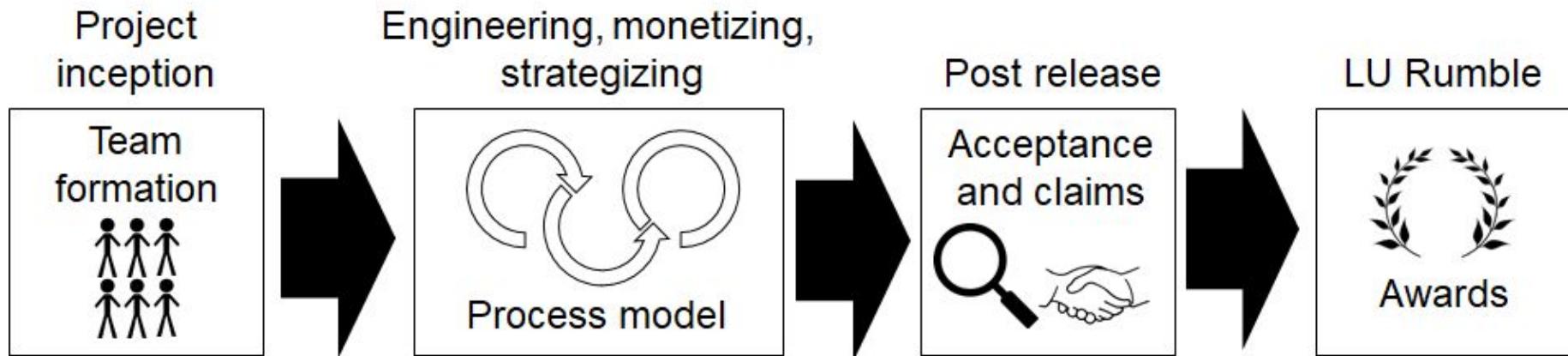
LUNDS
UNIVERSITET

Projektkontexten



LUND
UNIVERSITET

Projektfaser

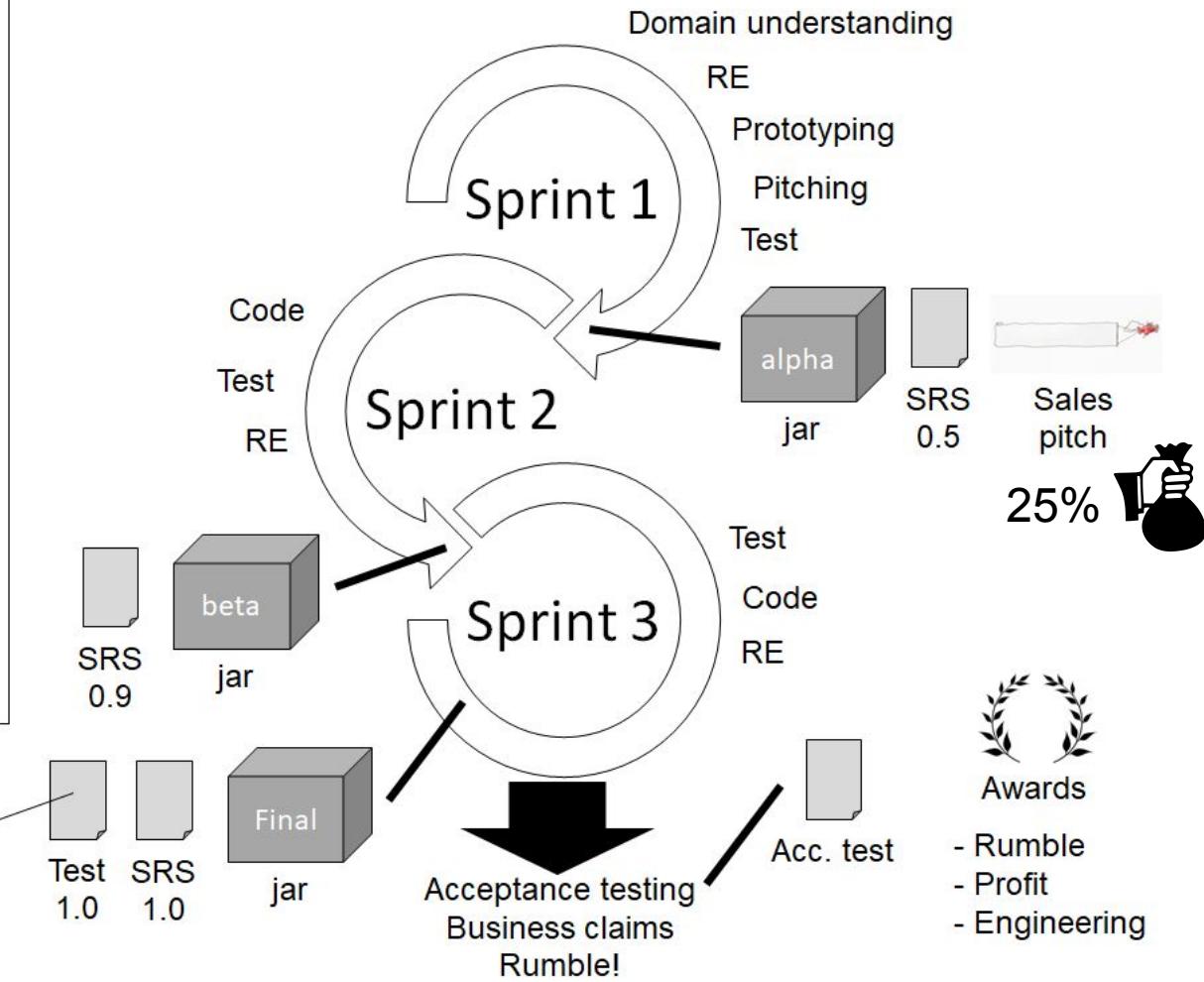


LUNDS
UNIVERSITET

Kärnan i projektet - tre sprints



75%
Test res



Projektuppgiften - leverantörsperspektiv

Utveckla och leverera en robot som ska användas av annan grupp

- jar-fil
- Kravspecifikation
- Specifikation för systemtest
- Testrapport

Den reguljära enheten vill även ha

- Källkod med god kvalitet
- Enhetstester



LUNDS
UNIVERSITET

Projektuppgiften - beställarperspektiv

Ta emot en robot från en annan grupp

- jar-fil
- Kravspecifikation
- Specifikation för systemtest
- Systemtestrapport

Utföra acceptanstest av robotten

- Granska kravspecifikation
- Acceptanstestrapport



LUNDS
UNIVERSITET

Projektuppgiften - relation och förhandling

Under Sprint 2 och Sprint 3

- Tät dialog mellan beställare och leverantör

Efter acceptanstest

- Beställare rapporterar till reguljära enheten
 - Allt ok! 100%  stannar har leverantören
 - Eller argumentera för ofullständig leverans
 $X\%$  kan återgå till beställaren

Er projekthandledare (som ingår i reguljära enheten) stöttar er!



LUNDS
UNIVERSITET

Projektuppgiften - Kommunikation

Slack

- “Searchable Log of All Conversation and Knowledge”
- Digitalt workspace för kommunikation
- Dominerande plattformen för utvecklarkommunikation

Huvudfunktioner

- Diskussionskanaler, publika och privata
- Direktmeddelanden
- Integrationer mot mängder av verktyg
- Användbara botar
- Öppna ekosystem med plug-ins





Projektuppgiften - Leverabel 1

Samarbetskontrakt och rollfördelning - senast torsdag kl. 23.59

- Lead developer: Huvudansvar för implementation av roboten. Gärna bekväm med programmering i Eclipse.
- Requirements engineer: Huvudansvar för att fånga kraven på roboten i en specifikation.
- Test lead: Huvudansvar för att verifiera att implementationen uppfyller kravspecifikationen.
- Domain expert: Huvudansvar för inköp av robot från annan grupp. Ska försöka skapa ett vinnande robotlag.
- Sales engineer: Huvudansvar för försäljning av gruppens robot. Skapar sales pitch och kommunicerar med domänexperter.
- Project manager: Huvudansvar för kommunikation med kursledningen. Rapporterar tid för teamets olika roller varje vecka.



LUNDS
UNIVERSITET

Projektuppgiften - osäkerhet

Första gången med helt nytt projekt

- Ny plattform
 - Okända tekniska utmaningar
- Införande av tävlingsmoment
 - Hur kommer LU Rumble att funka?
 - Hur kommer ETSA02s inre marknad att fungera?
- Ny interaktion mellan grupper
 - Bäddat för viss friktion

Räkna med att den reguljära enheten kommer att ändra på reglerna

- Osäkerhet är realistiskt för utvecklingsprojekt
 - Marknadens osäkerhet
 - Teknikens osäkerhet
 - Regelverkets osäkerhet



LUNDS
UNIVERSITET

Start: Etablera projektgrupper i pausen

~95 personer ⇒ 16 projektgrupper

Anmäl dig på anslagna lappar under pausen

Notera grupp 13-16 med handledning på engelska!

Kursledningen fördelar de som inte anmält sig

Gruppen ses senast på onsdag - datorövning 1

Grupp 1-8: kl. 13

Grupp 9-16: kl. 15



LUNDS
UNIVERSITET

Bilda projektgrupper

Skriv upp dig på en av grupperna.
Alla grupper ska ha sex deltagare.

Grupp XX	Efternamn, Förnamn	Program
on 13-15 E:3308	Karlsson, Kalle	C1
	Larsson, Lisa	I3

Grupper

1-8: Ons kl 13, Tors kl 8

9-16: Ons kl 15, Tors kl 10

- Notera grupp, tider och lokal



LUNDS
UNIVERSITET

Kravhantering



Kravhantering: Utveckla rätt produkt!



PicNic Pants by acquacalda

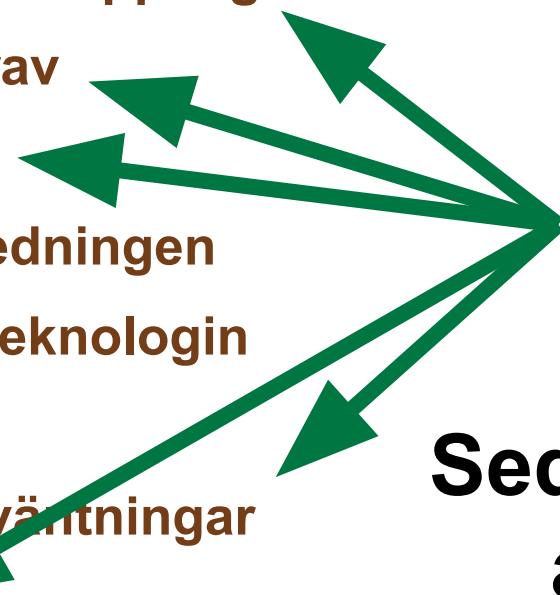
- Säkerställ att utvecklingen leder till önskad produkt
 - För marknaden?
 - Åt kontraktskund?
- Uttryck det som ska implementeras
- Designa ej lösningen!



LUNDS
UNIVERSITET

Software Engineering: Topp-10 utmaningar

1. Låg användaråterkoppling
2. Ofullständiga krav
3. Kraven ändras
4. Lågt stöd från ledningen
5. Behärskar inte teknologin
6. Resursbrist
7. Orealistiska förväntningar
8. Oklara mål
9. Orealistiska tidsplaner
10. Ny och obeprövad teknik



LUNDS
UNIVERSITET

Forskning tydlig: Krav avgörande!

- “Requirements Engineering and Downstream Software Development: Findings from a Case Study”, Damian et al., 2005
<http://link.springer.com/article/10.1007/s10664-005-1288-4>
- “Requirements Problems in Twelve Software Companies: an Empirical Analysis”, Hall et al., 2002
http://digital-library.theiet.org/content/journals/10.1049/ip-sen_20020694
- “Quantifying the Impact of Requirements Definition and Management Process Maturity on Project Outcome in Large Business Application Development”, Ellis and Berry, 2013
<http://link.springer.com/article/10.1007/s00766-012-0146-3>
- “Users' Involvement in Requirements Engineering and System Success”, Bano and Zowgi, 2013
<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6615212>

70 nya yrken v

■ Kravanalytiker, apputvecklare och akupressör är tre av de drygt sjuttio nya yrkesbenämningar som dykt upp i Arbetsförmedlingens register de senaste fem åren. Och på Folkuniversitetet kan man numera utbilda sig till eventkoordinator och visual merchandiser.

I och med digitaliseringen har till exempel yrket kravanalytiker dykt upp. Det innebär en person på ett företag som ansvarar för beställningen av digitala lösningar och ser till att det nya systemet är anpassat till verksamhetens behov. Samma sak gäller apputvecklare som tar fram appar till smarta telefoner. Hälstrenden världen över har lett till en rad nya jobb inom den sektorn. Ett av

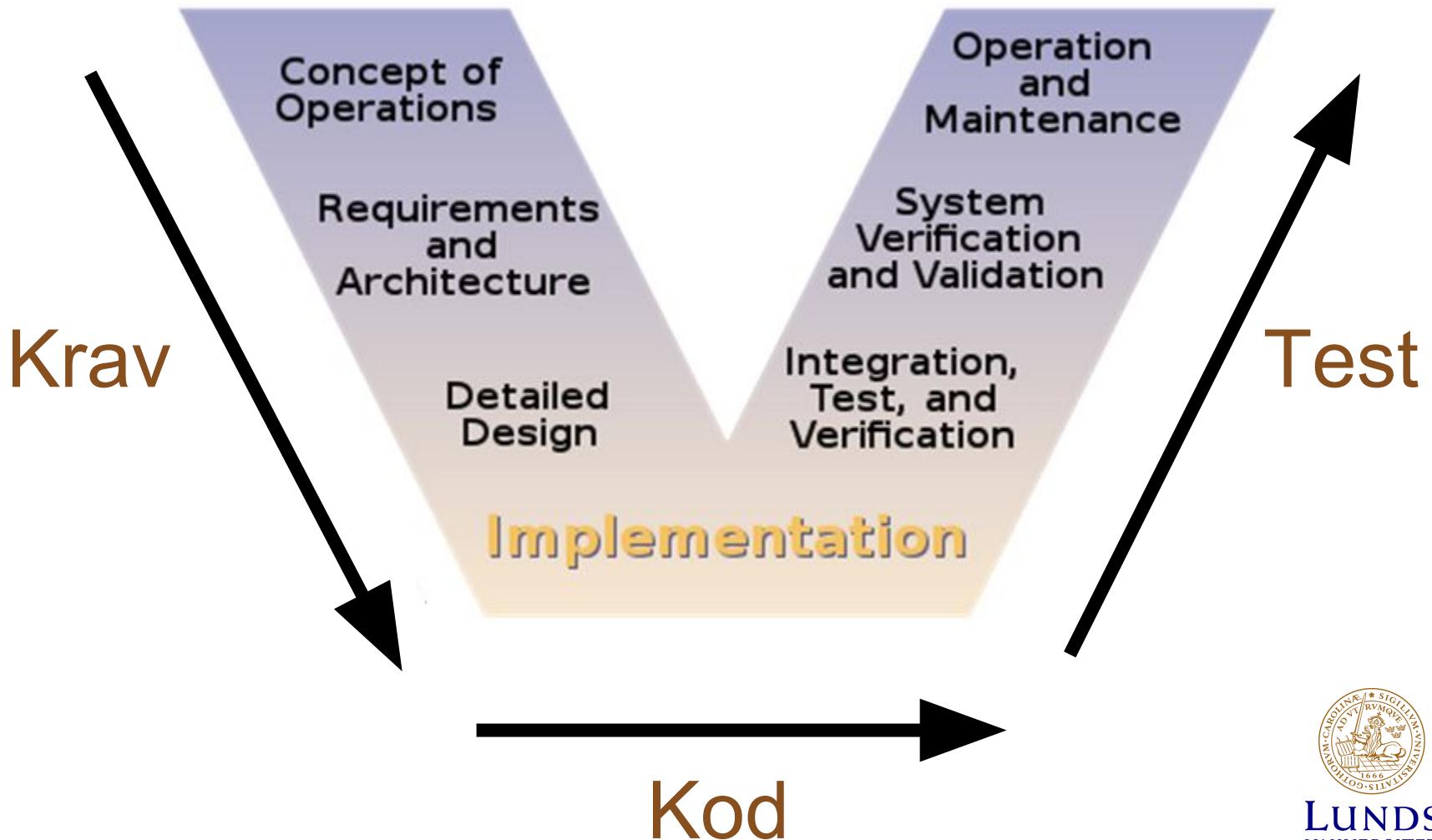


Vad är ett krav?



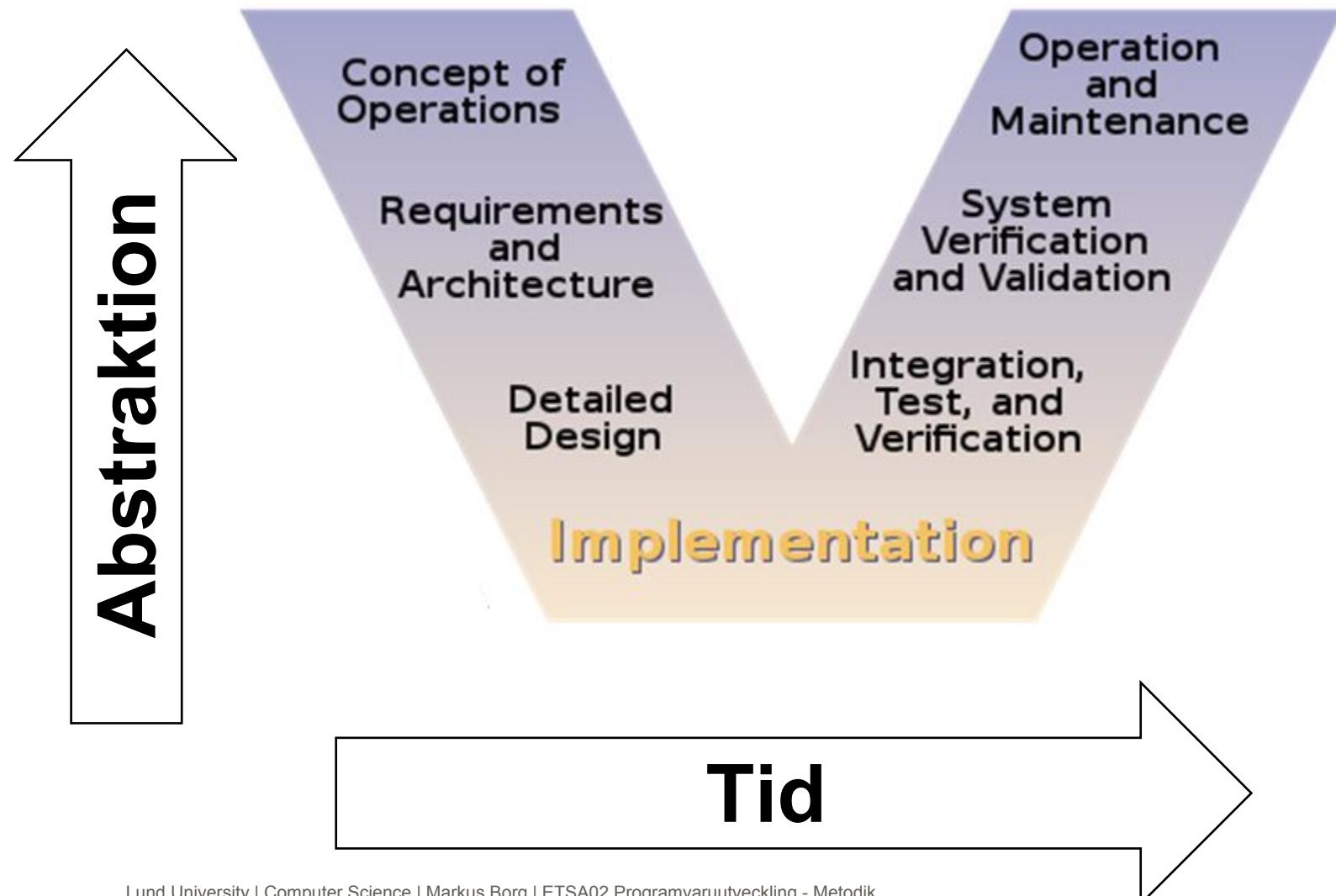
LUNDS
UNIVERSITET

Större perspektiv: V-modellen för systemutveckling



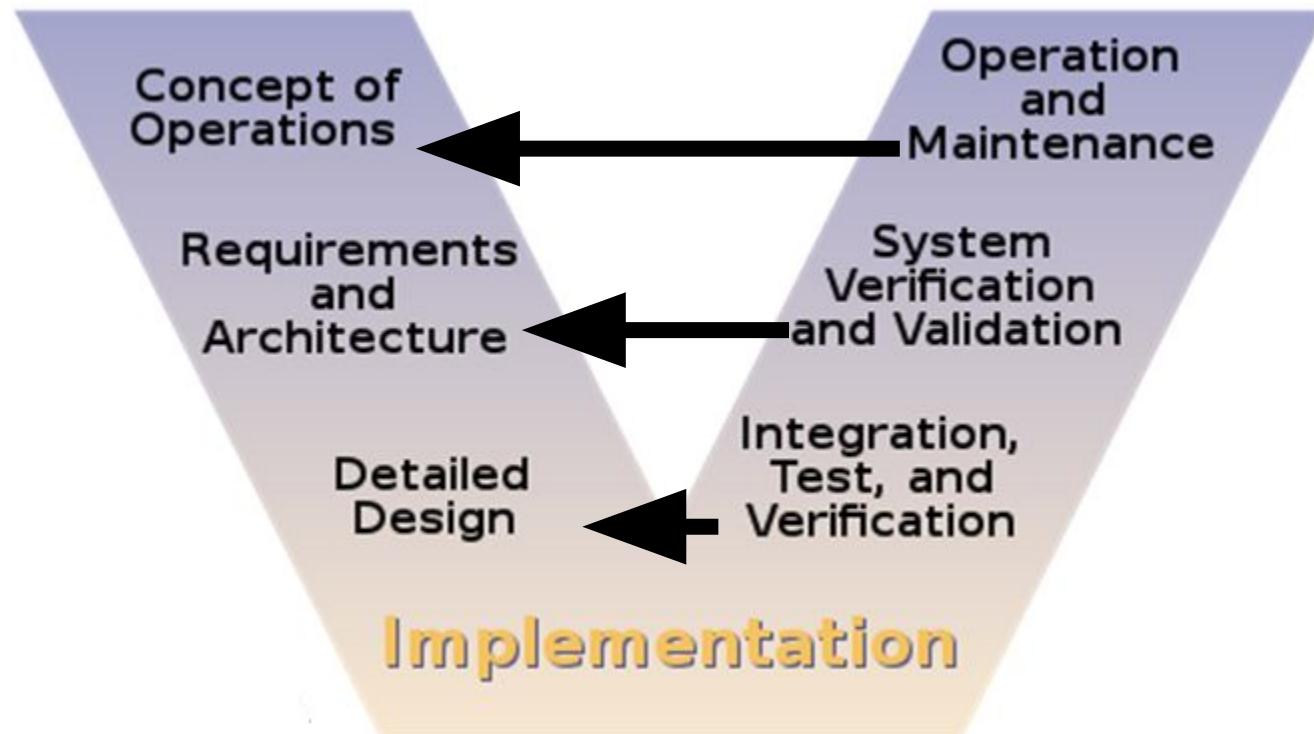
LUNDS
UNIVERSITET

V-modellen: Dimensioner



LUNDS
UNIVERSITET

V-modellen: Koppling Krav-Test



Vi forskar på dessa kopplingar i Lund!
"Challenges and Practices in Aligning Requirements with Verification and Validation: A Case Study of Six Companies"
Bjarnason et al., 2014, <http://link.springer.com/article/10.1007/s10664-013-9263-y>

Två huvudtyper av krav

Funktionella krav

- Beskriver vilka funktioner systemet ska erbjuda

Kvalitetskrav

- Exempelvis begränsningar för funktionerna
- Påverkar ofta hela produkten

“Icke-funktionella krav”

Tumregel

- Funktionella krav binära
 - uppfylls / uppfylls ej
- Kvalitetskrav behandlas på en glidande skala
 - t.ex. långsam, snabb, snabbare, snabbast



**LUNDS
UNIVERSITET**

Kvalitetskrav – olika kategorier

Tillförlitlighet

- Mognadsgrad, feltolerans, återhämtningsförmåga

Användbarhet

- Begriplighet, lärbarhet, handhavande, attraktivitet

Effektivitet

- Tidsbeteende, resursutnyttjande

Underhållbarhet

- Analyserbarhet, ändringsbarhet, stabilitet, testbarhet

Portabilitet (flera plattformar)

Uppfyllandegrad (standarder etc)

Exempel: Läskautomat



Flickr: hibino



LUNDS
UNIVERSITET

Läskautomat: Funktionella krav

Om kunden erlägger belopp större än en varas pris ska systemet returnera mellanskillnaden.

Vid time-out returnerar systemet erlagda mynt.



**LUNDS
UNIVERSITET**

Läskautomat: Kvalitetskrav

Det får maximalt gå 1.0 sekund från en myntläggning till att systemet är redo att ta emot nästa mynt.

Programvaran får högt använda 65 kb ROM.

Systemet får vara ur funktion högst 30 minuter om året.



LUNDS
UNIVERSITET

Kravhanteringsprocessen

“Process” enligt Wikipedia ≈

En samling i förväg uttänkta aktiviteter som ska användas varje gång man skapar ett visst resultat.

Identifiera krav

Analysera krav

**Dokumentera
krav**

Validera krav



**LUNDS
UNIVERSITET**

1. Identifiera krav

Från olika personer med olika behov

Ta hänsyn till lagar, regler och standarder

Hur?

- Marknadsanalyser
- Kundkontakter
- Analys av befintliga system
- Intervjuer
- Kartläggningar (frågeformulär etc)
- Prototyper
- Brainstorming



LUND
UNIVERSITET

2. Analysera krav

- På djupet förstå problemen som systemet ska lösa
- Leder till att:
 - nya krav identifieras
 - krav stryks
 - krav förändras
- Särskilt viktigt att tidigt:
 - Korrekta, dvs. stämmer med bakomliggande behov
 - Kompletta, dvs. inga viktiga saker saknas



LUNDS
UNIVERSITET

3. Dokumentera krav

- Utveckla kravspecifikation för systemet
- Olika format möjliga
 - Text i naturligt språk (vanligast!)
 - Grafiska format
 - Formella metoder
 - » exempelvis matematiska uttryck och modellbaserad utveckling



LUNDS
UNIVERSITET

Naturligt språk: ”Skall-krav”

SRS-B-42 *The system shall support 100 simultaneous users.*

Fördelar:

- Generellt
- Flexibelt
- Lätt att använda och förstå

Nackdelar:

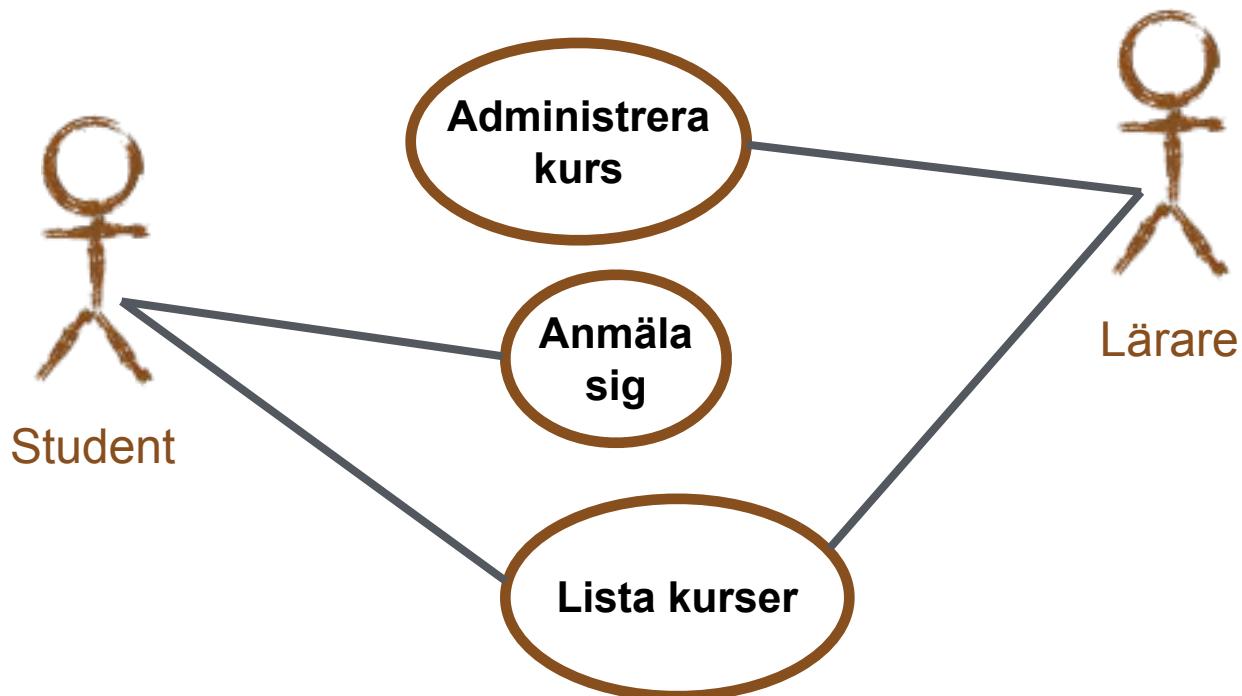
- Otydligt
- Tvetydigt
- Risk att skapa motsättningar
- Risk att slå samman flera krav i ett



LUNDS
UNIVERSITET

Användningsfall (use case)

Identifiera, analysera, dokumentera och validera krav, utgående från typiska exempel på användning.



LUNDS
UNIVERSITET

Användningsfall - exempel anmäla till kurs

Primär aktör: Student

Förhandsvillkor: Studenten är inloggad

Huvudscenario

1. Studenten listar tillgängliga kurser
2. Studenten klickar på knappen "Anmäl" vid en listad kurs
3. Systemet registrerar valet och hälsar studenten välkommen

Undantagsfall

3a) Studenten uppfyller ej förkunskapskrav
=> Systemet meddelar studenten "Uppfyller ej förkunskaperna"

3b) Kursen är fulltecknad, studenten får inte plats
=> Systemet placerar studenten i kö och meddelar studenten:
"Kursen för närvarande full, du har placerats i kö"



LUND
UNIVERSITET

Skapa användningsfall - Metod

Identifiera aktörer och deras mål

För varje användningsfall

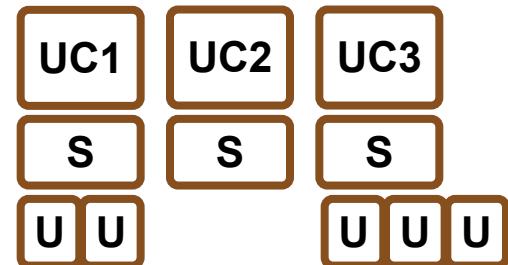
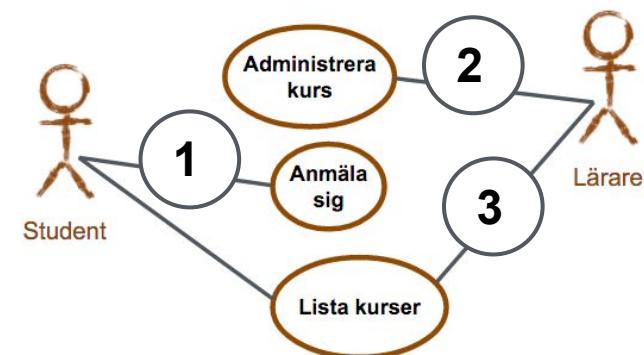
- förstå och specificera huvudscenario
- beskriv förhandsvillkor

För varje huvudscenario

- identifiera undantagsscenarier

För varje undantagsfall

- specificera vad som ska hända



LUNDS
UNIVERSITET

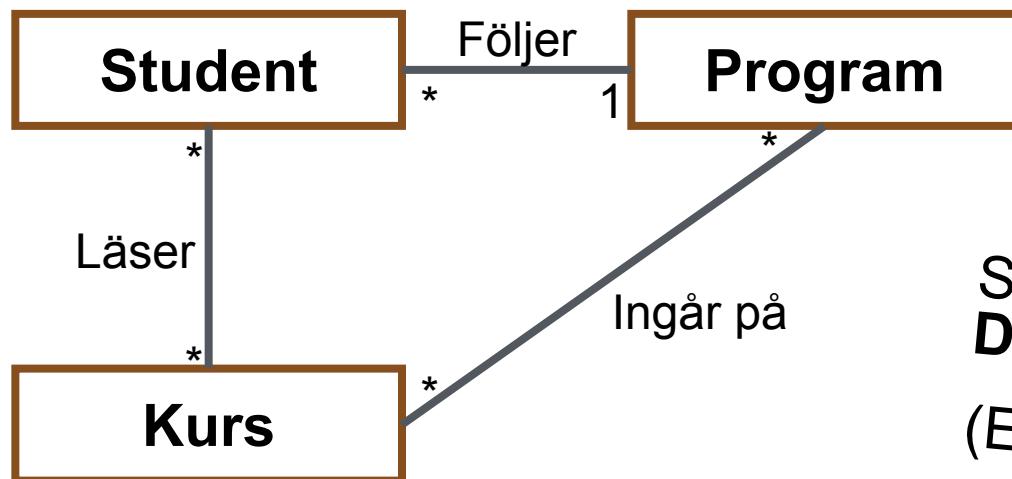
Grafiska format: ER-diagram

Statisk beskrivning av data (Entity Relationship)

Student: personnummer, namn, inskrivningsår

Kurs: kurskod, namn, #högskolepoäng

Program: beteckning, namn



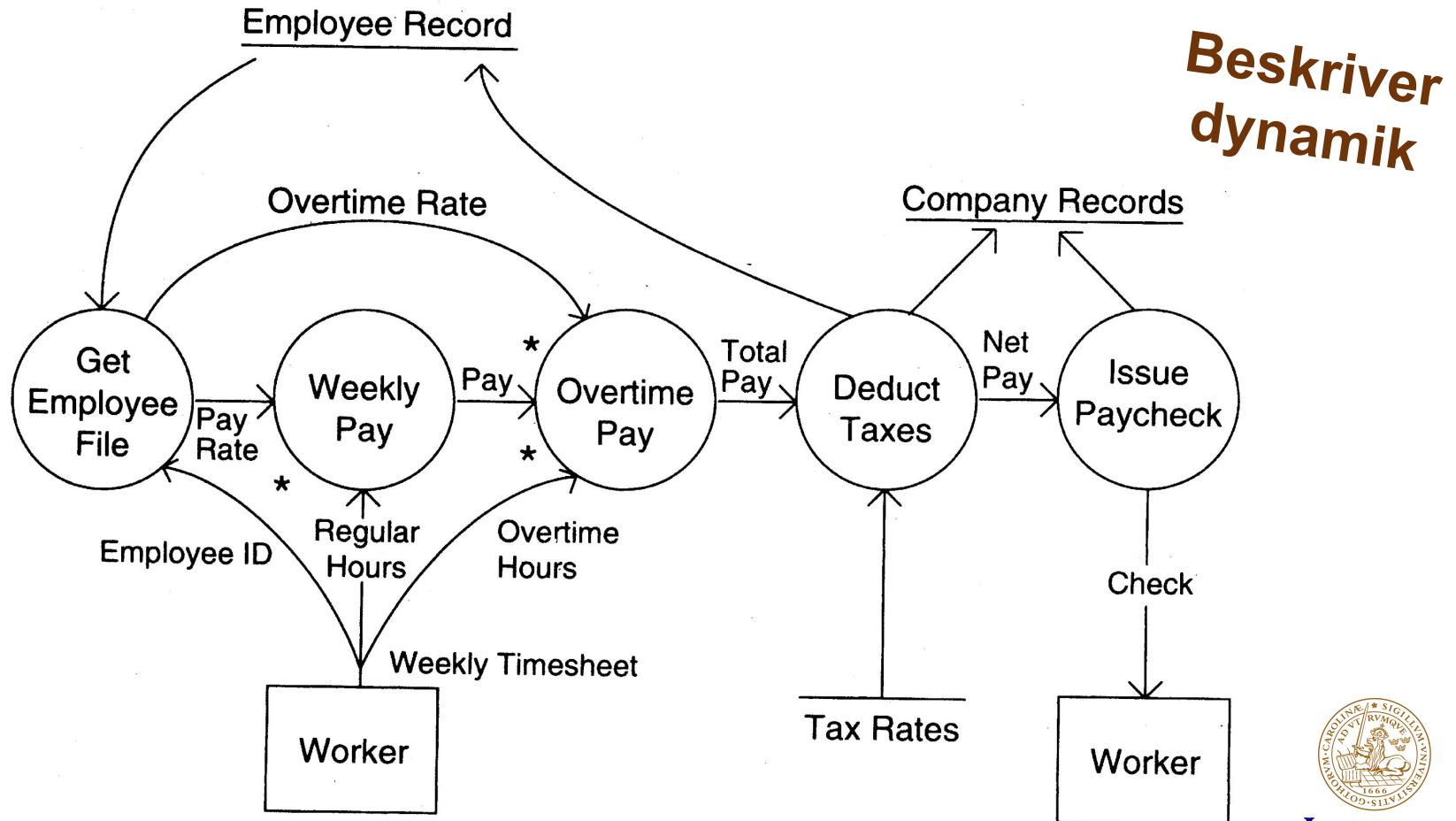
Används ofta för
att modellera
data i databaser

Se även kursen
Databasteknik
(EDAF75)



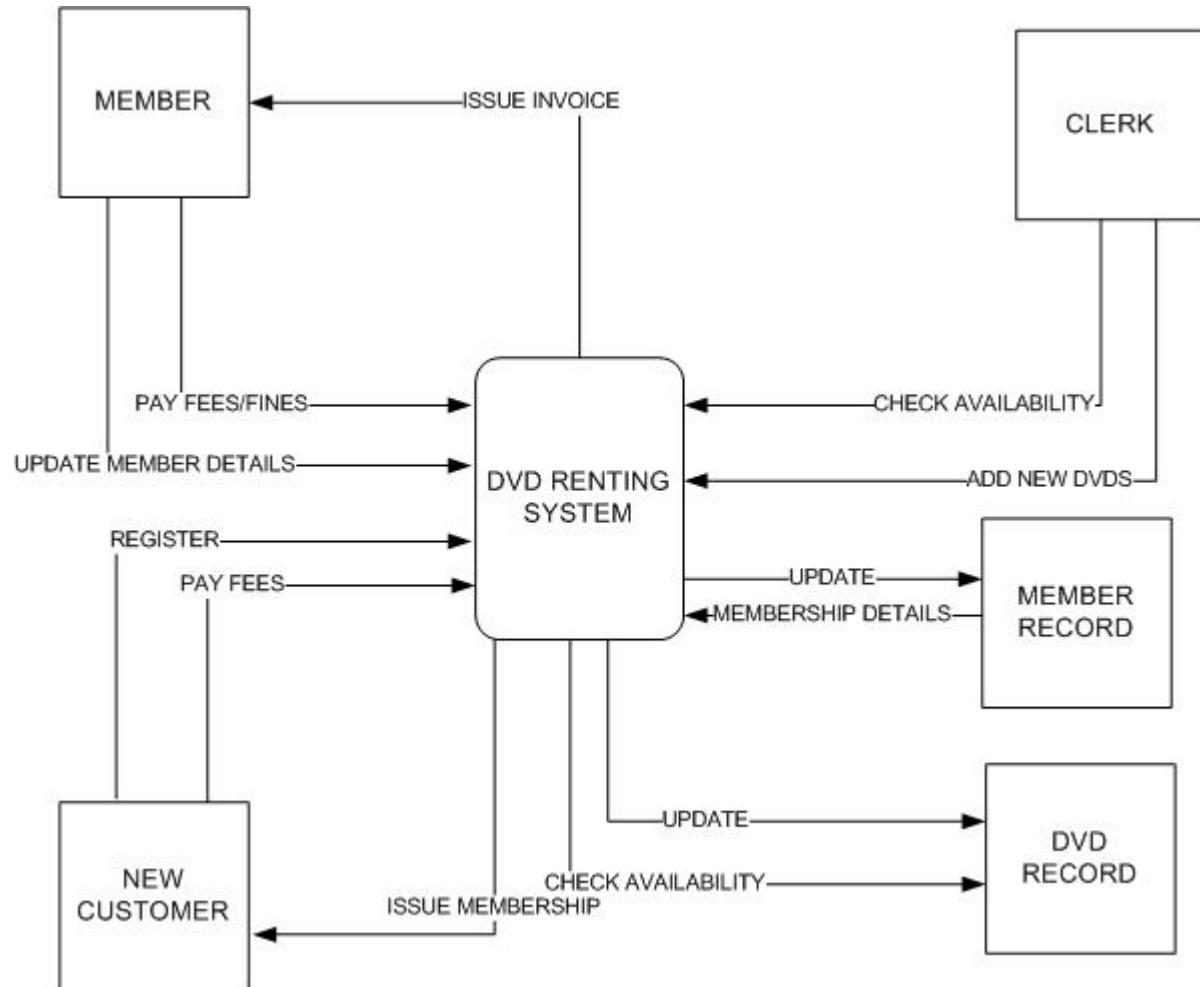
LUNDS
UNIVERSITET

Grafiska format: Dataflödesdiagram



LUNDS
UNIVERSITET

Grafiska format: Kontextdiagram



LUND
UNIVERSITET

Formella metoder: Z-notation

project : OZSpec → UMLDiagram

$\forall(oz, uml) : project \bullet$

$\{c : oz \cap Classdef \bullet c.name\} = \{c : uml.classes \bullet c.name\} \bullet \forall c_1, c_2 : oz \cap Classdef \bullet \exists_1 c' : uml.classes \bullet c'.name = c_1.name$

$c'.attrs = \{cls : Classdef \mid cls \in oz \bullet cls.name\}$
 $\triangleleft c_1.state.decpart$

$c'.ops = \{o : Opdef \mid o \in c_1.ops \bullet o.name\}$

$c_2.name \in \{t : \text{ran } c_1.state.decpart \bullet t.name\} \Rightarrow$
 $\exists_1 (c'_1, c'_2) : uml.agg \bullet c'_1.name = c_1.name$
 $\wedge c'_2.name = c_2.name$

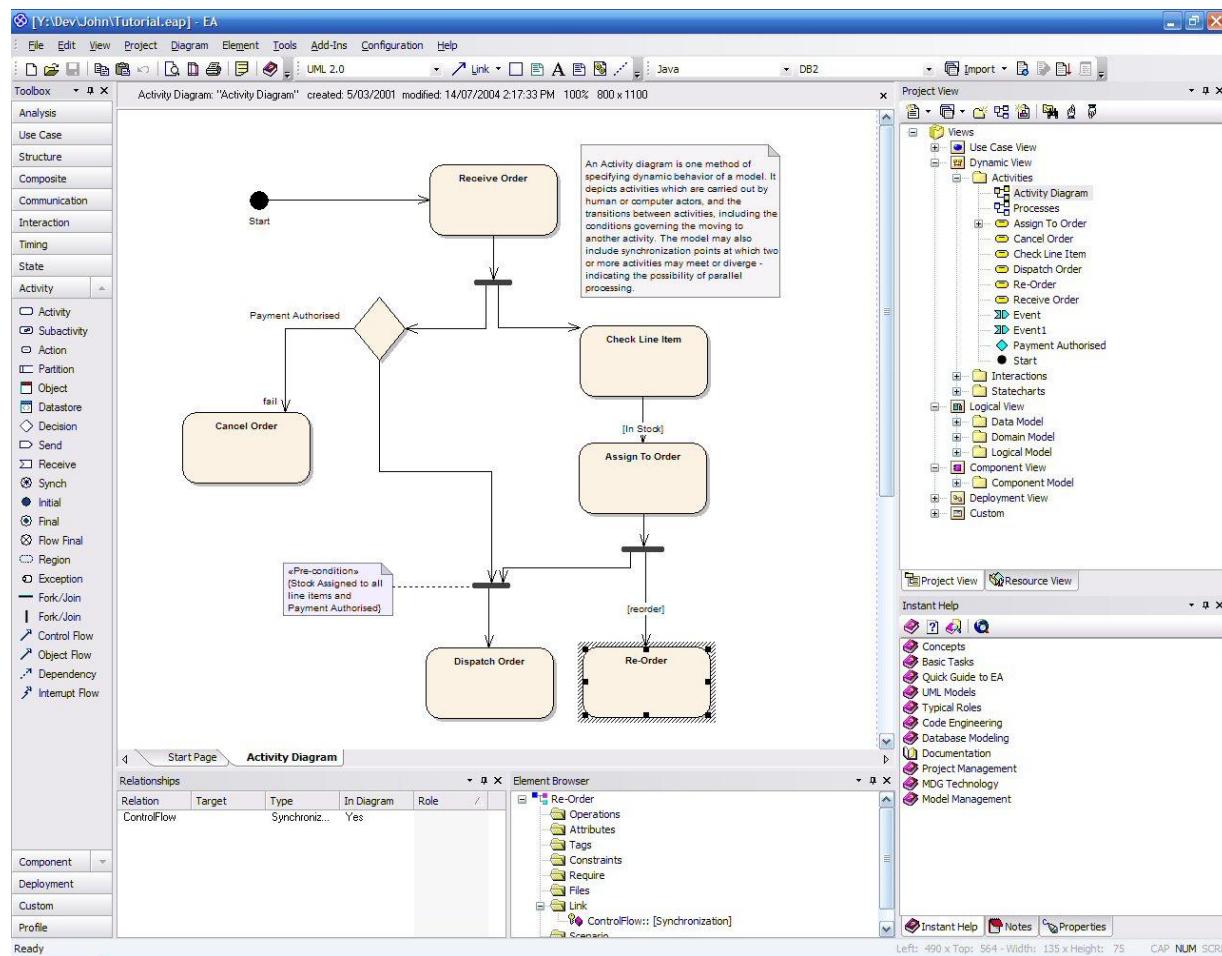
$c_2.name \in \{inh : \text{dom } c_1.inherit \bullet inh.name\} \Rightarrow$
 $\exists_1 (c'_1, c'_2) : uml.inh \bullet c'_1.name = c_1.name$
 $\wedge c'_2.name = c_2.name$

Möjliggör
matematisk
bevisföring



LUNDS
UNIVERSITET

Formella metoder: Modell-baserad utv.



Abstrahera förbi
högnivåspråk

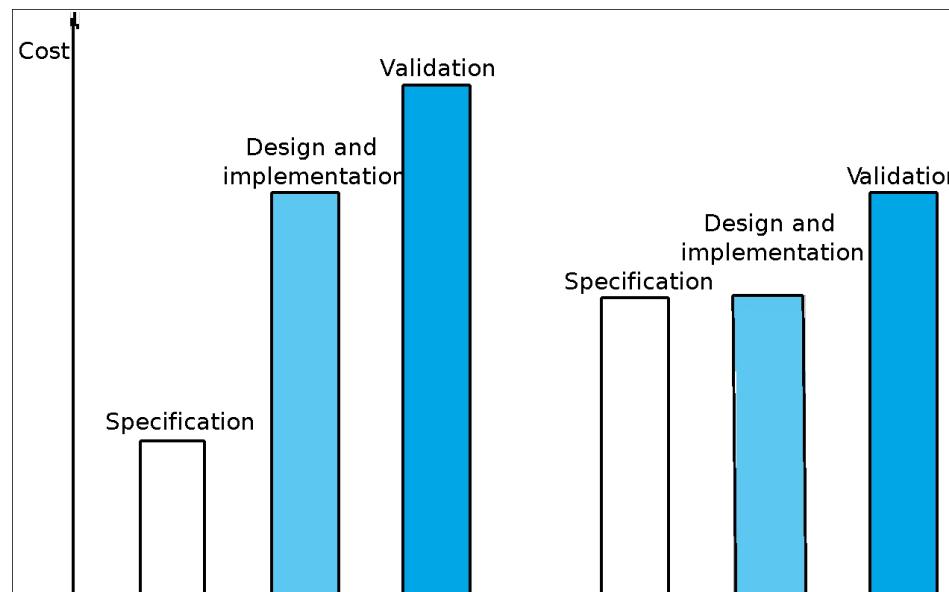
1. Uttryck krav i modelleringsverktyg
2. Generera kod



LUND
UNIVERSITET

Formella metoder: Summering

- **Mer arbete med specifikation \Rightarrow mindre jobb med implementation och verifiering**
- **Begränsad spridning i industrin – flyg, fordon, kärnkraft etc.**



(Sommerville, 2004)



LUNDS
UNIVERSITET

Struktur i kravspecifikation

Table of contents

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms,
and Abbreviations

1.4 References

1.5 Overview

2. General Description

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

2.4 General Constraints

2.5 Assumptions and
Dependencies

3. Specific Requirements

Appendix

Index

[IEEE Guide to Software
Requirements Specifications,
ANSI/IEEE Std 830-1984]



LUNDS
UNIVERSITET

Exempel: Dokument från myndigheter

Bred begäran av allmänna handlingar
har gjorts 2017-01-01

- Teknisk dokumentation från mjukvaruutvecklingsprojekt
- Samtliga svenska förvaltningsmyndigheter har kontaktats (~250)
- Exempel kommer att finnas på Google Drive



LUNDS
UNIVERSITET

4. Validera kraven

Kontrollera att kravspecifikationen är korrekt och av hög kvalitet

Exempel på metoder:

- Granskning (vanligast)
- Bjud in till fokusgrupper
- Utveckla prototyper med viss funktionalitet



**LUNDS
UNIVERSITET**

Bra egenskaper hos krav och en kravspecifikation

- Korrekt
- Heltäckande
- Otvetydigt
- Konsistent
- Verifierbart
- Nödvändigt
- Spårbart
- Givet prioritet



Vid time-out
returnerar systemet
erlagda mynt.

Korrekt	✓
Heltäckande	-
Otvetydigt	✓
Konsistent	-
Verifierbart	✓
Nödvändigt	✓
Spårbart	-
Rankat	-



LUNDS
UNIVERSITET

Checklista för granskning i 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 alla texter och illustrationer nödvändiga?
12. Har samtliga krav unika identifierare?



LUNDS
UNIVERSITET

Kravprioritering

Omöjligt att implementera alla bra idéer!

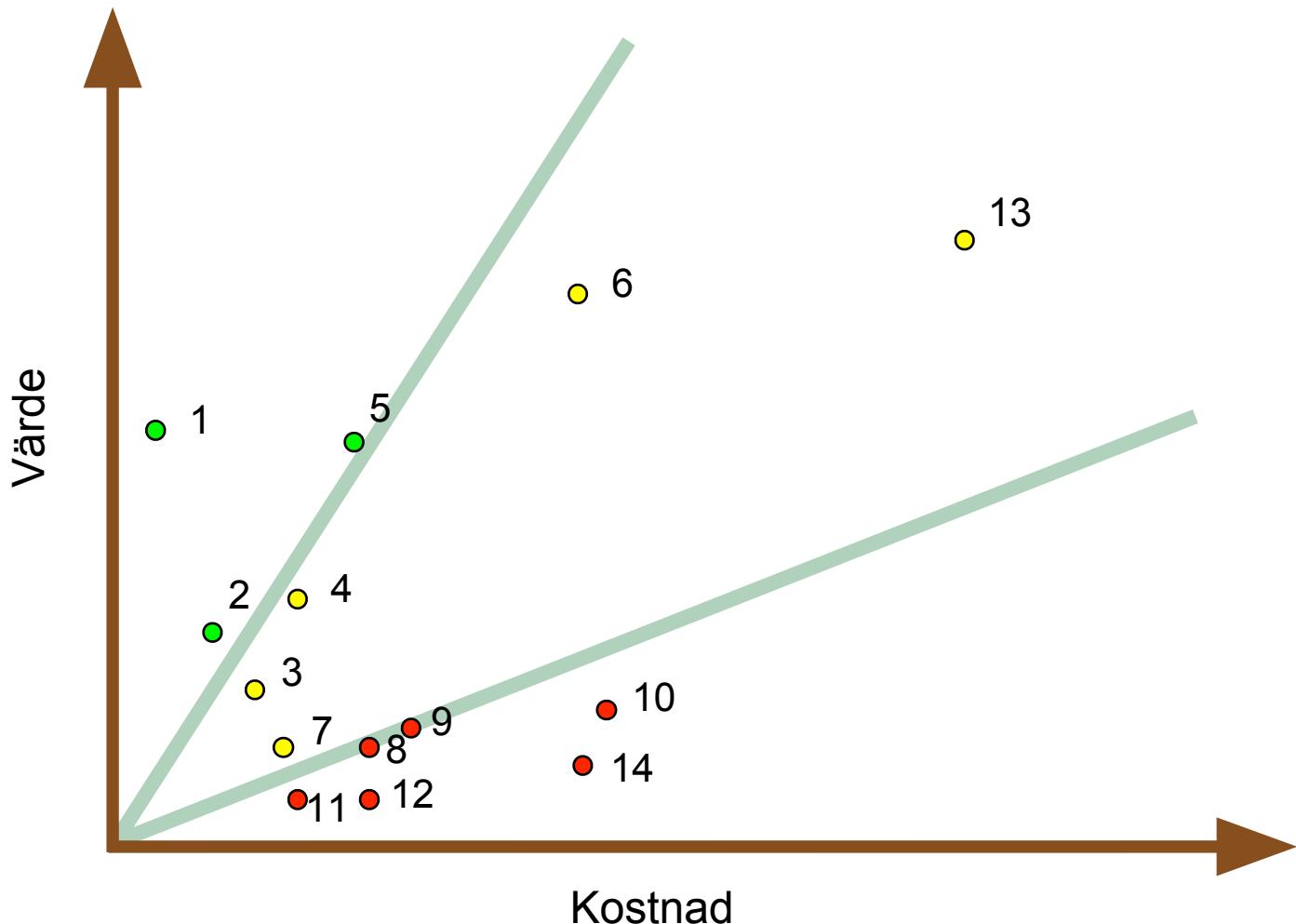
Jämför krav med avseende på:

- Värde för kund
- Kostnad att implementera
- Ledtid att implementera
- Risk
- Säljargument i marknadsföring
- Påverkan på varumärket/produktportfölj
- Utgångspunkt för vidareutveckling



LUNDS
UNIVERSITET

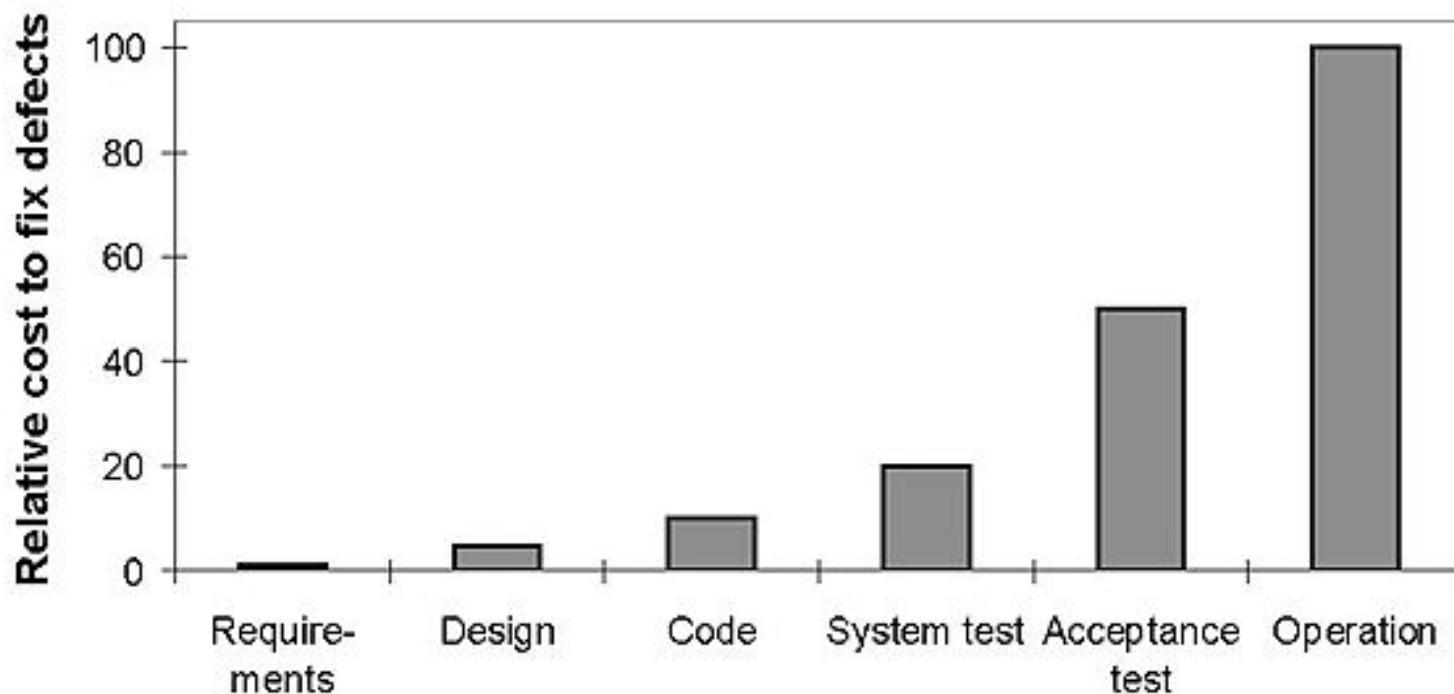
Exempel prioritering - värde/kostnad



LUNDS
UNIVERSITET

Kostnader för felrättning skenar

Ju senare ett fel hittas, desto dyrare att fixa!



**LUNDS
UNIVERSITET**

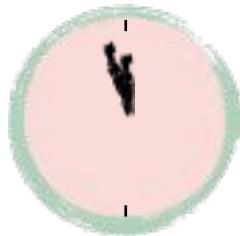
Sammanfattning - Krav

- Utan genomtänkt målbild ➔ stor risk för misslyckat projekt
- Viktigt eftersom tidiga projektfaser påverkar mycket
- Krav kan finnas på olika abstraktionsnivåer
 - Visioner, features, detaljerade krav
- Kvalitetskrav påverkar ofta hela produkten
- Egenskaper hos krav: korrekt, komplett, otvetydigt, nödvändigt, verifierbart, konsistent, prioriterat, spårbart

Vi forskar på kravhantering i Lund!
Fördjupning ges i kursen **Kravhantering** (ETSN15)



LUNDS
UNIVERSITET



Att göra nu!

Prio 1: Kom på banan!

- Läsa in er på projektbeskrivningen
- Kolla imorgon att ni kommer in på projektwebben och Slack
- Diskutera rollfördelning

På onsdag träffas vi i datorsal.

Då startar projektet.
Var där!

Prio 2:

- Läs om kravhantering i boken
- Titta på samarbetskontraktet (L1: Deadline Torsdag 23:59)

Drive

SKAPA

Min enhet

Delade med mig

Stjärnmarknad

Senaste

Papperskorgen

Mer ▾

RUBRIK

ETSA01 Grupp 01

ETSA01 Alla

ETSA01 Mallar och

Dra projektmappen från **Delade med mig** till **Min enhet**

