



LUNDS
UNIVERSITET

ETSA01 Ingenjörprocessen för Programvaruutveckling – Metodik

Föreläsning 1 – Markus Borg





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?



5

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

Markus Borg - CV



Civilingenjör Datateknik, Lund

2002-2007

- Programvaruutveckling
- Exjobb reglerteknik/datavetenskap



Utvecklingsingenjör ABB, Malmö

2007-2010

- Processautomation



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



Markus Borg - CV



Civilingenjör Datateknik, Lund

2002-2007



LUNDS
UNIVERSITET

- Programvaruutveckling
- Exjobb reglerteknik/datavetenskap

Utvecklingsingenjör ABB, Malmö

2007-2010

- Processautomation
- Säkerhetskritisk utveckling
- Utveckling av kompilatorer och editorer



Doktorand Programvarusystem, Lund

2010-2015



LUNDS
UNIVERSITET

- Forskning kring bugggrätning i stora system
- Spårbarhet och projektintern sökning



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

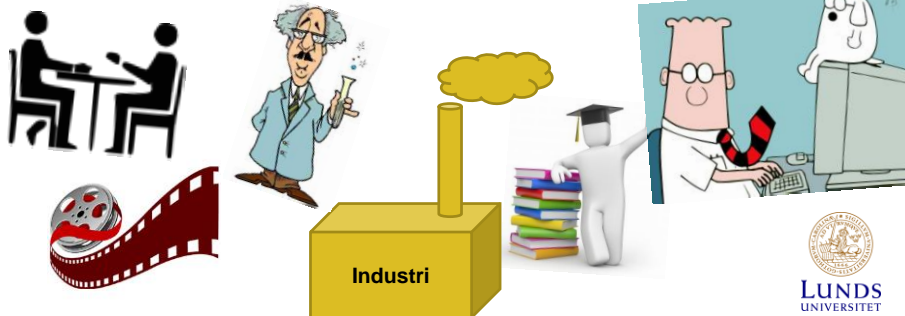
Agenda föreläsning 1

Allmänt om kursen

Projektuppgiften

Kravhantering

I pausen: Bilda
projektgrupper och
köpa kompendier



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



Allmänt om kursen

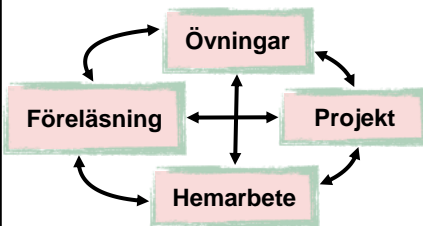
Kursen

Innehåll

- Kravhantering
- Projektplanering
- Arkitektur
- Testning
- Processmodeller

Formalia

- 5 hp
- Obligatorisk för C1, D1, alternativobligatorisk för I3
- Moment
 - 6 Föreläsningar
 - 4 Övningar (två dubbla)
 - Projekt
 - Hemtentamen
- Första delen av kurskedja
 - Soft issues + Economy and Quality



Kurslitteratur

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

– J: 6.2-5, 7.1.1-7.1.3 kursivt



2. Kompendium
Examples and Exercises in the Software Engineering Process, 2014.

– Säljs av CS-institutionen för 50;-



V	måndag					tisdag					onsdag					torsdag					fredag				
	8	10	12	13	15	8	10	12	13	15	8	10	12	13	15	8	10	12	13	15	8	10	12	13	15
13																									
14																									
15																									
16																									
17																									
18																									
19																									
20																									
21																									
22																									
23																									
24																									
25																									

Livscykel från projektet		Planerad återkoppling från projekthanterarna	
	L1 Användarförståelse samt funktionskrav och kvalitetskrav	->	fb1 Att gruppen har förstått grunderna i formulering av användarfall och krav.
	L2 Kravspecifikation 1.0 av 0.2	->	fb2 Att kravspecifikationen motsvarar enligt plan.
G1 Granskning av Kravspecifikation 0.90 inför 0.99	L3 Kravspecifikation 0.99, Granskningsprotokoll från granskning inför Kravspecifikation 0.99, Projektplan med mål och risker	->	fb3 Att krav, granskningsprotokoll och projektplan är av tillräcklig kvalitet för att minsta 1 ska kunna passeras.
G2 Granskning inför Kravspecifikation 1.0	L4 Kravspecifikation 1.0, granskningsprotokoll	->	fb4 Kravspecifikation OK eller ytterligare iteration
G3 Granskning inför Testplan 1.0 och Design 1.0	L5a Testplan 1.0, Design 1.0		
G4 Extern QA-granskning av testplan 1.0	L5b Extern QA-granskning		
	L6 Slutskrivelse av kravspec, Projektplan, Testplan, Design, mötesnotiser löd, testprotokoll, manual för systemstart	->	fb5 Granskning och godkännande av samtliga dokument
		->	fbT Tentamenresultat i LADOK.
Föreläsningar	Föreläsningar	Övningstfällen	Föreläsningar
F1 Kursöversikt, kravhantering, gruppindelning, projektuppgiften, intro till Ö 1	1: 1, 2, 4	Ö1a	Söker på plats
F2 Kravhantering, Projektplanering, Granskning.	3: 3, 4, 7, 5	Ö1a	Projektkravstyrningen avsnitt 1-4 på kurswebben. A: R.1-6
F3 Test, konfigurationer och design	3: 8	FW	Introduktion till projekt R: 7-10
F4 Design och praktisk testning	3: 5, 6, 7		
F5 Utvecklingsprocesser, vidareutveckling, om tentamen	3: 2	Ö1b	Gemensam diskussion om gruppen användarfall, Diskussion: A: R.1-6 Göra A: L2-3 (fr senaste R2)
F6 Inför tentamen, sammanfattning av kursen	3: 1-8, AHE1	Ö2	A: R.1-5 A: I.1
		Ö3	A: T.1-8 Diskussion: A: T.1-8 Göra: A: T.11-12 Måttlig R2
		Ö4a	A: T.11-14 Diskussion: A: T.11-14 Göra: A: T.9-10 Introduktion till projektets inför A4b
		FW	
		Ö4b	Uppföljning FW gemensam diskussion kring projektets testfall Förstärkt testplanering

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

LUNDS
UNIVERSITET

Examination

A. Projektarbete – 3 hp

B. 26 timmar hemtenta - 2 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

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

Personal

Markus Borg

- kursansvarig, föreläsningar, övningar

Jonas Wisbrant

- föreläsning, övning

Anna Axelsson

- övningar, projekthandledning

Sandra Nilsson

- övningar, projekthandledning

Jesper Holmén Notander

- projekthandledning

Sardar Muhammad Sulaman

- projekthandledning

Hoa Troung

- projekthandledning

Josefin Karlsson

- projekthandledning

Lisa Stenström

- projekthandledning

Jacob Arvidsson

- projekthandledning

Olof Wahlgren

- projekthandledning

Carolina Sartorius

- projekthandledning



LUNDS
UNIVERSITET

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

Hälsningar från olika håll...

Förra årets 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 – nytexaminerade studenter idag är mycket bättre förberedda för projektarbete än för 10 år sedan



LUNDS
UNIVERSITET

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



LUNDS
UNIVERSITET

Om projektuppgiften



Projektuppgiften - storn

Utveckla och leverera programvara till ett garage som förvarar cyklar och hanterar användare

- Applikation
- Installationsmanual

Arbetsgivaren och beställare vill ha insyn i ert projekt:

- Projektplan
- Granskningsprotokoll
- Testrapporter från systemtest

Ert projekt ska kunna vidareutvecklas av någon annan:

- Kravspecifikation
- Testspecifikation
- Designdokument
- Källkod med god kvalitet



I praktiken

Arbeta i grupper om 6 personer

Genomför ett litet utvecklingsprojekt från början till slut

- Kravidentifiering och kravanalys
- Projektplanering
- Design
- Implementation
- Testning
- Leverans

Leverabler

- Kravspecifikation
- Projektplan
- Testplan & testspecifikation
- Granskningsrapporter
- Designdokument
- Manual
- Testrapporter
- Exekverbar applikation

Plattformer

- Projektwebb för dokumenten
- Java/swing för programvaran

Obligatorisk dokumentstruktur i projektwebben

ETSA01 Grupp XX

DOKUMENT Förteckning över projektets leveranser (med länkar till dokumenten)

DOKUMENT: Projektplan

DOKUMENT: Installationsmanual

MAPP: 1 Kravspecifikation

DOKUMENT: Kravspecifikation 0.1 - Usecase 1 inför Ö1b och L1

DOKUMENT: Kravspecifikation 0.2 - L2 Sanity Check

DOKUMENT: Kravspecifikation 0.9 - För intern granskning på väg mot 0.99

DOKUMENT: Kravspecifikation 0.99 - L3 För extern granskning

DOKUMENT: Kravspecifikation 1.0 - L4 Milstolpe 1

MAPP: 2 Testplan

DOKUMENT: Testplan 1.0 - L5 Milstolpe 1 - för extern granskning

MAPP: 3 Designdokument

DOKUMENT: Design 1.0 - L5 Milstolpe 1 - för extern Protokoll för extern granskning

MAPP: Externa granskningsprotokoll

MAPP: Interna granskningsprotokoll

DOKUMENT: Granskningsprotokoll för granskning av Kravspecifikation 0.90

DOKUMENT: Granskningsprotokoll för granskning av design 0.90



LUNDS
UNIVERSITET

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

Arbetsbelastning - budgetidé för 133 h

Läsa kursbok	20h
Hemtenta	10h
Föreläsningar	12h
Individuella övningsförberedelser	6h
Övningstillfällen	12h
Obligatoriska granskningsmöten	6h
Delsumma	66h

Exempel projektschema för projektgrupp 1 för vecka 1-3



Projekttid	67h
Summa	133h



~ 12 h / läsvecka



LUNDS
UNIVERSITET

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

Kick-start: Etablera projektgrupper i pausen

201 personer --> 33 projektgrupper

Anmäl dig på anslagna lappar under pausen

Kursledningen fördelar de som inte anmält sig

Grupp 1-15: ONSDAG KL 13

Grupp 16-33: ONSDAG KL 15



LUNDS
UNIVERSITET

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

Bilda projektgrupper och köp kompendium i pausen

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

Grupp XX	Deltagare (namn)	Program
on 13-15 E:3308	Lisa Larsson	D1
	Kalle Karlsson	I3

- Notera grupp, tider och lokal
- Köp kompendium

Grupp 1-15						
	Onsdag				Torsdag	
V	8	10	13	15	8	10
13			X	x	X	
15					X	
16					X	
17			X	x	X	

Grupp 11-30						
	Onsdag				Torsdag	
V	8	10	13	15	8	10
13				X	x	X
15						X
16						X
17				X	x	X

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



Kravhantering: Utveckla rätt produkt!



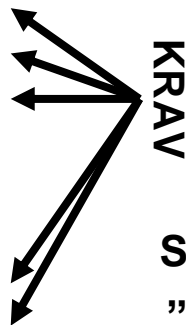
PicNic Pants by acquacalda

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



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

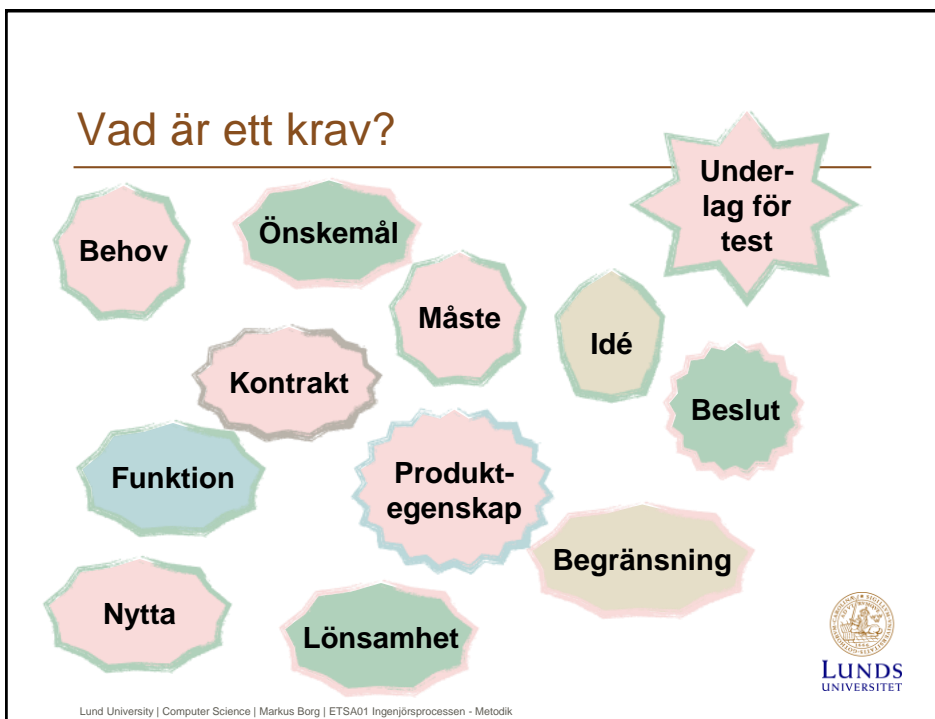


**Sedan 1994 årliga
"Chaos Reports"**

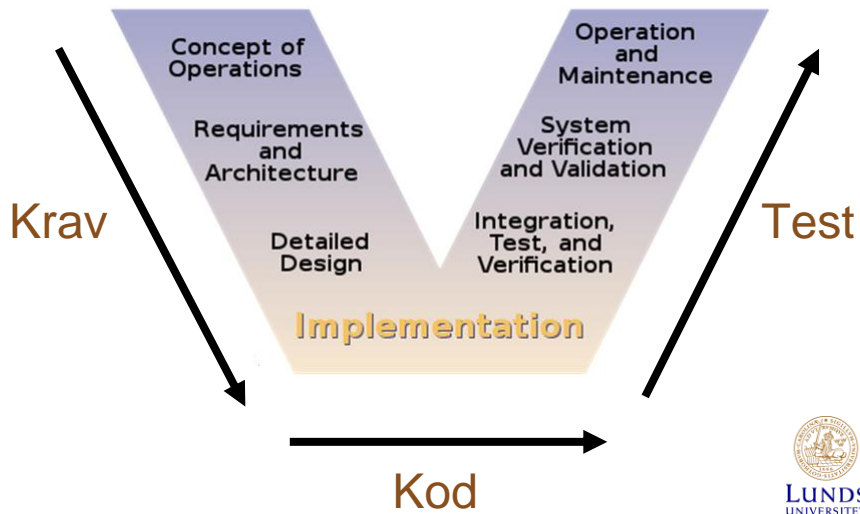


Forskning tydlig: Krav avgörande!

- "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>

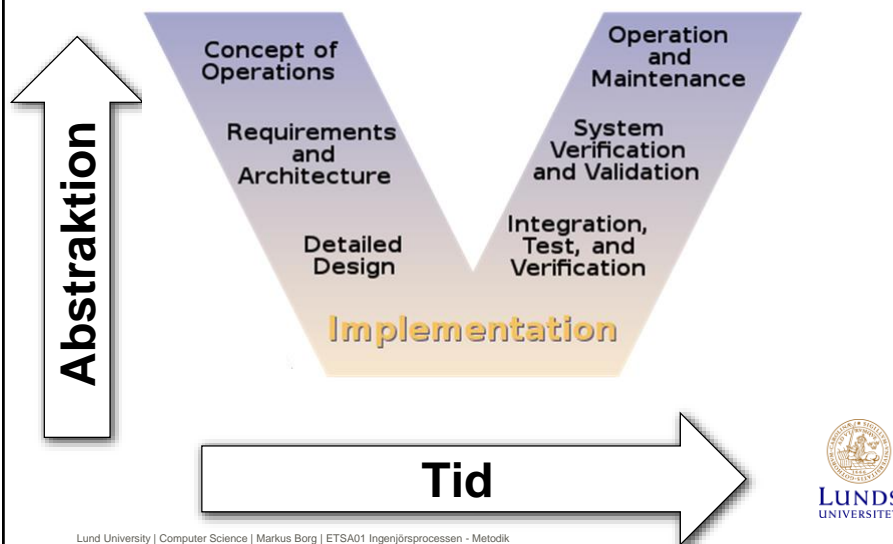


Större perspektiv: V-modellen för systemutveckling



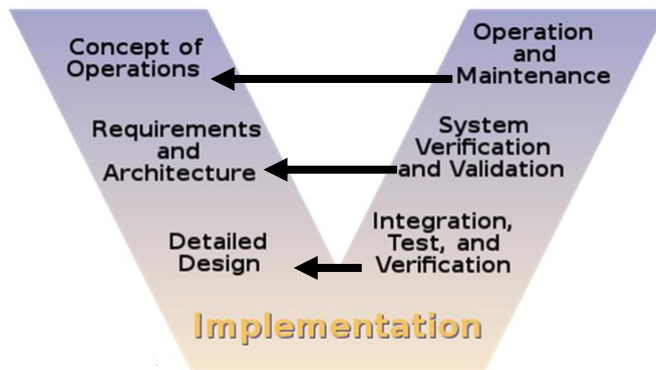
Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörprocessen - Metodik

V-modellen: Dimensioner



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

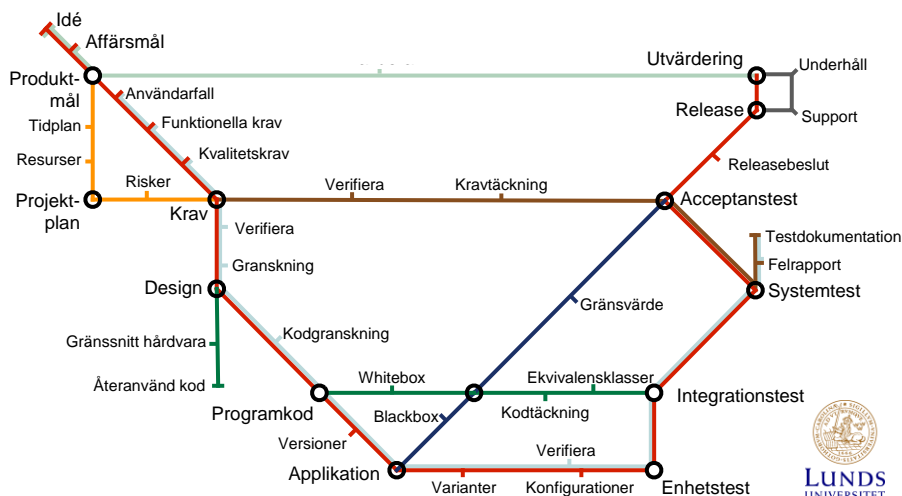
V-modellen: Koppling Krav-Test



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>

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

Mental modell: Tunnelbana

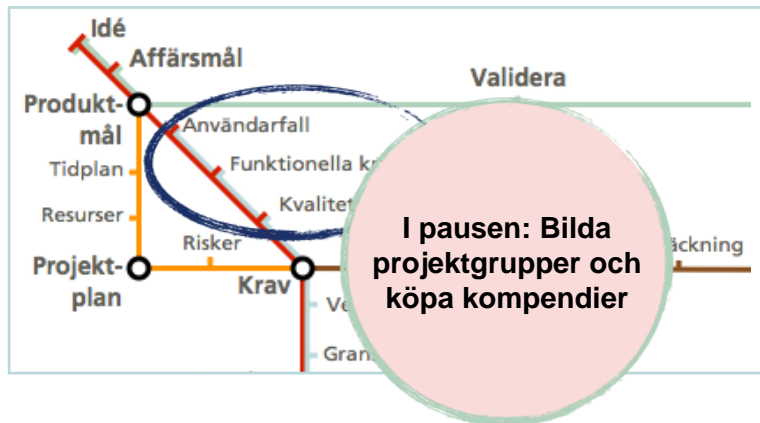


Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörprocessen - Metodik



LUNDS
UNIVERSITET

Dagens föreläsning...



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



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

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



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)

ISO9126 ingår i Ingenjörprocessen 3
– Ekonomi och kvalitet

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

Exempel: Läskautomat



LUNDS
UNIVERSITET

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



Läskautomat: Funktionella krav

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

Vid time-out returnerar systemet erlagda mynt.

Om en kund trycker på en knapp för en vara som inte finns händer ingenting.



Läskautomat: Kvalitetskrav

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

Programvaran får högst använda 65 kb ROM.

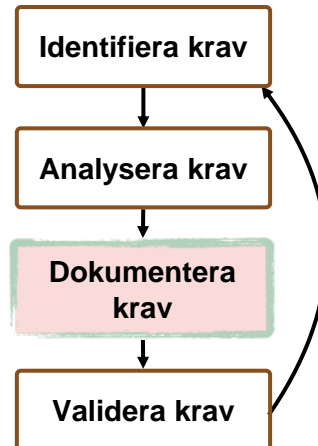
Systemet får vara ur funktion högst 30 minuter om året.



Kravhanteringsprocessen

En Process enligt Wikipedia ≈

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



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

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



2. Analysera krav



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



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

I projekten



Naturligt språk: "Skall-krav"

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

Fördelar:

- Generellt
- Flexibelt
- Lätt att använda och förstå

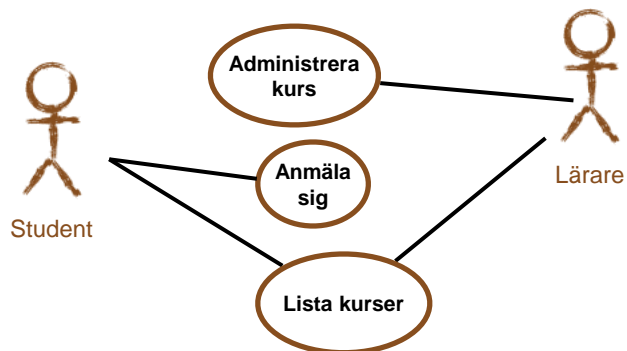
Nackdelar:

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



Användningsfall (use case)

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



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"



Skapa användningsfall - Metod

Identifiera aktörer och deras mål

För varje användningsfall:

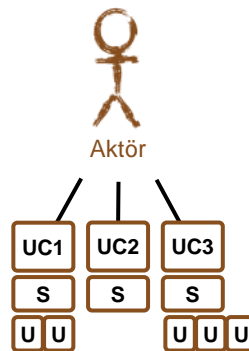
- förstå och specificera huvudscenario
- beskriv förhandsvillkor

För varje huvudscenario

- identifiera undantagsscenarier

För varje undantagsfall:

- specificera vad som ska hända



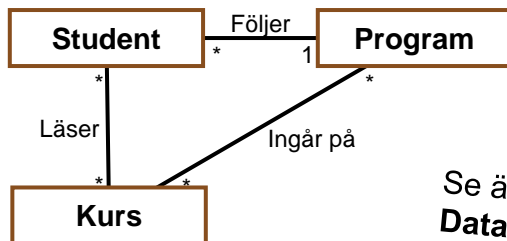
Grafiska format: ER-diagram

Statisk beskrivning av data (Entity Relationship)

Student: pnr, namn, inskrivningsår

Kurs: kurskod, namn, #hp

Program: beteckning, namn



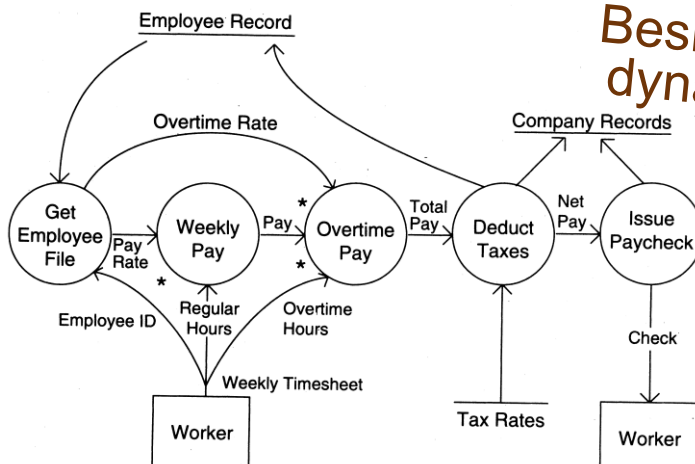
Används ofta för
att modellera
data i databaser

Se även kursen
Databasteknik
(EDA216)

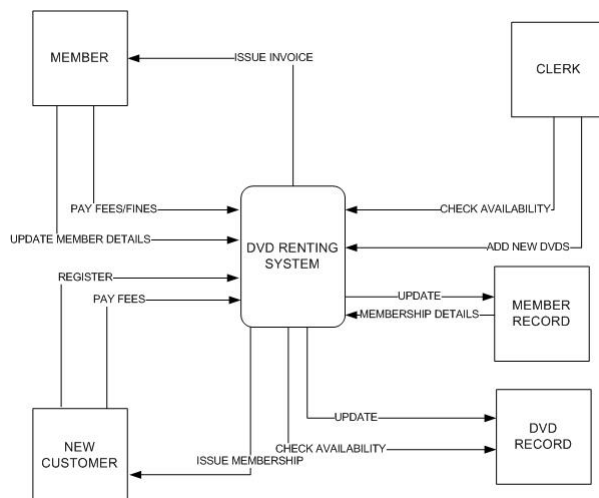


Grafiska format: Dataflödesdiagram

Beskriver
dynamik



Grafiska format: Kontextdiagram



Måste finnas i
projektens
kravspecar!



LUNDS
UNIVERSITET

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

Formella metoder: Z-notation

$project : OZSpec \rightarrow UMLDiagram$

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

$\{c : oz \cap Classdef \bullet c.name\} = \{c : uml.classes$

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

$uml.classes \bullet c'.name = c_1.name$

$c'.attris = \{cls : Classdef \mid cls \in oz \bullet cls.name\}$

$\triangleleft c_1.state.decpart$

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

$c_2.name \in \{t : ran c_1.state.decpart \bullet t.name\} \Rightarrow$

$\exists_1(c'_1, c'_2) : uml.agg \bullet c'_1.name = c_1.name$

$\wedge c'_2.name = c_2.name$

$c_2.name \in \{inh : dom c_1.inherit \bullet inh.name\} \Rightarrow$

$\exists_1(c'_1, c'_2) : uml.inh \bullet c'_1.name = c_1.name$

$\wedge c'_2.name = c_2.name$

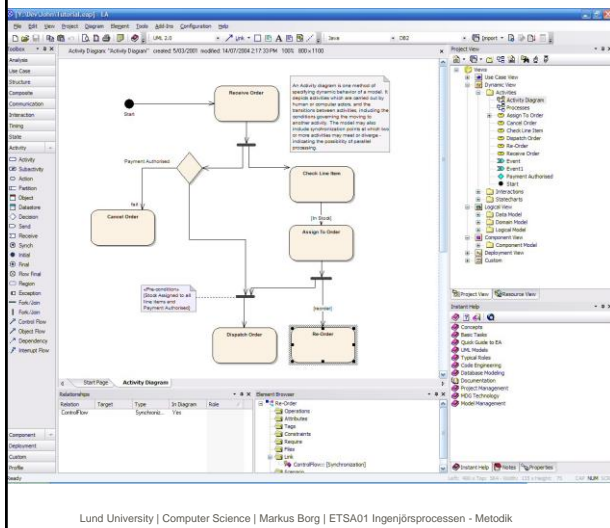
Möjliggör
matematisk
bevisföring



LUNDS
UNIVERSITET

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

Formella metoder: Modell-baserad utv.



Abstrahera förbi högnivåspråk

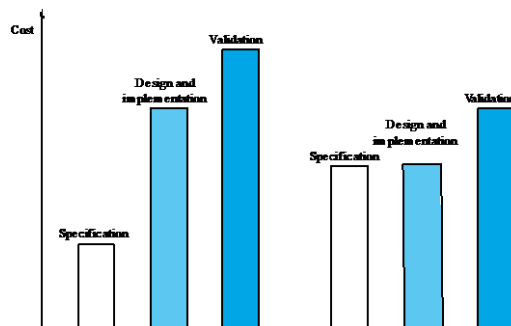
1. Uttryck krav i modellingsverktyg
2. Generera kod



LUND
UNIVERSITET

Formella metoder: Summering

- Mer arbete med specifikation => mindre jobb med implementation och verifiering
- Begränsad spridning i industrin (flyg, fordon, kärnkraft etc.)



(Sommerville, 2004)



LUND
UNIVERSITET

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

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]



Exempel: Kravspecar från verkligheten

Mentcare system

- Informationssystem för hantering av patientdata
- Patientsäkerhet viktigt
- Domineras av skall-krav

<http://iansommerville.com/software-engineering-book/files/2014/10/Mentcare-requirements-document.pdf>

KeePass Password Safe

- Databas för hantering av lösenord (öppen källkod)
- Kryptering fundamentalt
- Domineras av användningsfall

<http://keepass.info/extensions/v1/docs/SoftwareRequirementsSpecification-KeePass-1.10.pdf>



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



Bra egenskaper hos krav och en kravspecifikation



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

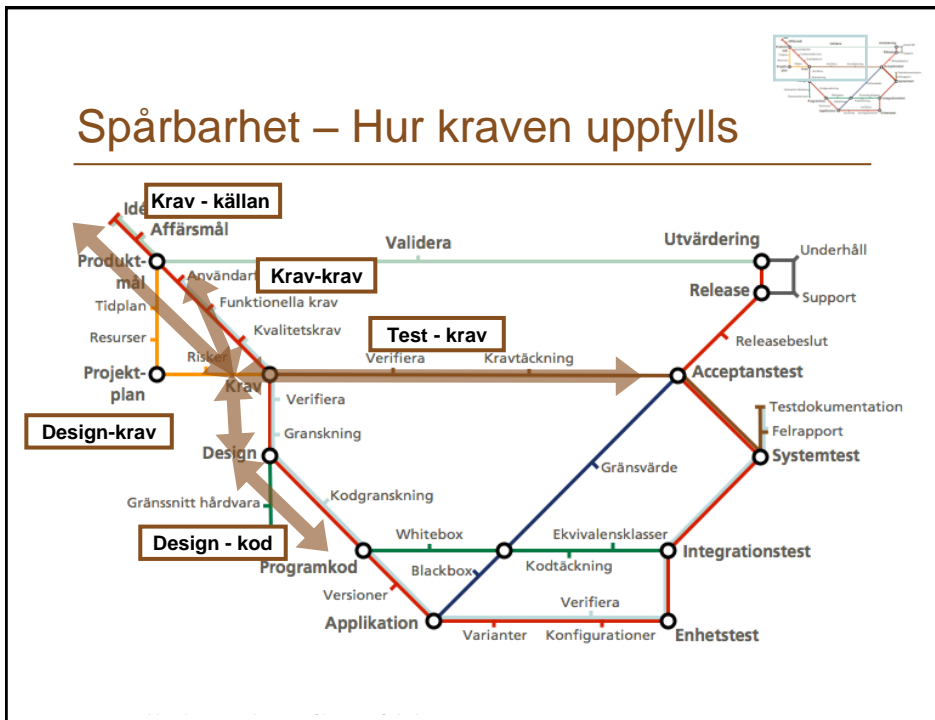


Vid time-out
returnerar
systemet
erlagda mynt.

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



Spårbarhet – Hur kraven uppfylls



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 alla texter och illustrationer nödvändiga?
12. Har samtliga krav unika identifierare?

Kravprioritering



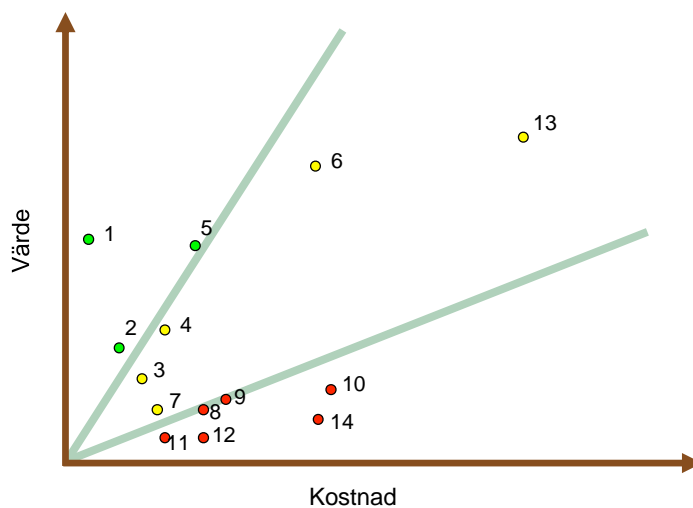
Omöjligt att implementera alla bra idéer!

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

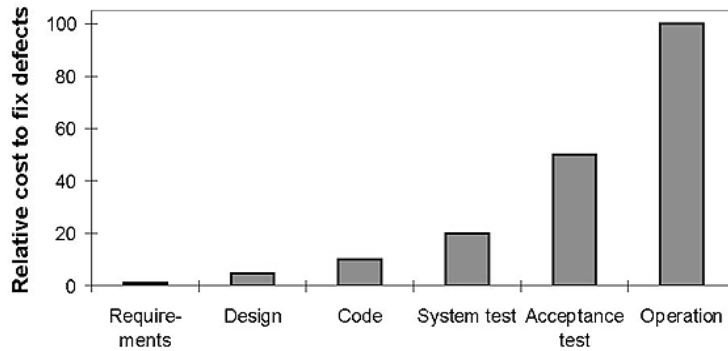


Exempel prioritering - värde/kostnad



Kostnader för felrättning skenar

Ju senare ett fel hittas, desto dyrare att fixa!



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



Finns alltid en Dilbert...



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



Sammanfattning - Krav



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

Vi forskar på kravhantering i Lund!
Fördjupning ges i kursen **Kravhantering** (ETS170)



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

Att göra nu!



Prio 1: Kom på banan!

- Läsa in er på projektbeskrivningen
- Göra R.1-6
- Kolla att ni kommer in på projektwebben

Prio 2:

- Läsa J:3-4

Denna vecka

Ö1a: Mer om krav + intro till Ö1b

PW: Schemalagt projektarbete

Fre kl 24: L1 i projektwebben

Nästa vecka

Föreläsning 2:

- Mer om kursen
 - Om projektplanering
 - Om granskning
 - Val av kursombud
- Ö2: Om projektplanering och dokumentgranskning

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