



LUNDS
UNIVERSITET

Föreläsning 6: Summering och om tentamen

ETSA01 Ingenjörprocessen 1 - Metodik VT15 | Markus Borg



Agenda

Kursinformation

Sammanfattning av kursen och operativ utvärdering

- Schemalagda kursaktiviteter
- Cykelgarageprojektet

Tentamen

- Så går det till – hemtentaprocessen!
- Genomgång av tidigare tenta

Kursavslutning och framtida kurser



Kursinformation

Nu är det vecka 20 -> slutinlämning om 2 veckor + 13,5 h

V 20:

- Må kl 10 F: Sammanfattning, utvärdering, om tentamen
- Mån kl 23.59: L5b - Extern QA-granskning

V 22:

- Må kl 23.59: L6 – Slutinlämning

V 23:

- Må kl 08 – Ti kl 10 - Hemtentamen

Kring midsommar:

- Återkoppling på projekt och projektbetyg
- Resultat från hemtentan



Manual för systemstart

Hur gör jag för att:

- installera systemet?
- få igång det?
- logga in?
- börja använda det?

Se kurswebb för recept på att exportera stand alone applikation

Recept för att skapa exekverbar jar-fil från eclipse

När skripten lämnas in är det tänkt att den ska gå att starta genom att man i princip dubbelklickar på den. Det som installatören behöver veta är att inte vara beroende av att ha Eclipse eller annan utvecklingsplattform installerad. För att skapa en exekverbar JAR-fil i Eclipse gör du så här:

1. Kör programmet i Eclipse. Då skapas en s.k. konfiguration. Märks på önskat projekt, projektmål, filgenmäkta och Next-As och lämpligt Run-konfigurationerna så ser du vilka konfigurationer som skapats.
2. Filgenmäkta vad som helst i projektyrket och välj Export -> Ett nytt körbart (Export) -> Java.
3. Klicka på ja när dialogen öppnas och sedan på Run. Detta skapar en JAR-fil i mappen Run i projektet. Detta är den JAR-fil som ska användas för att starta programmet.
4. Välj Önskad konfiguration i projektmålet i Eclipse. Ange målbildning och önskat framvar (som ska ha tillägg .jar) i läst Export-destination. Klicka på Färdig.

Under förutsättning att Java Runtime Environment (JRE) är installerad bör det nu skapas en .jar-fil i mappen Run. Detta är så att köra genom att man dubbelklickar på den.

© 2014 Markus Borg

Målgrupp:

Systemadministratör
≠ operatör...





LUNDS
UNIVERSITET

Snabbrepris och utvärdering av enskilda kursmoment

ETSA01 Ingenjörprocessen 1 - Metodik VT15 | Markus Borg




Snabbspolande utvärdering av kursen

Exempel sida 1:

Schemalagda kursmoment

Deltog

F1. Kursöversikt, kravhantering, gruppindelning, projektuppgiften, Intro till övning 1			Sal	
Positivt Bra överblick	Negativt För tidigt i kursen	Nya idéer Håll föreläsning på engelska istället	X	

Exempel sida 2:

Cykelgarageprojektet

Del-
tog

Läro-
rikt 1-
10

Kravhantering				
Positivt	Negativt Svårt logga in på Google-kontot	Nya idéer Fler konkreta exempel	X	8

Lämna anonymt i lådorna vid utgången

Glöm inte att besvara CEQ-enkäten!

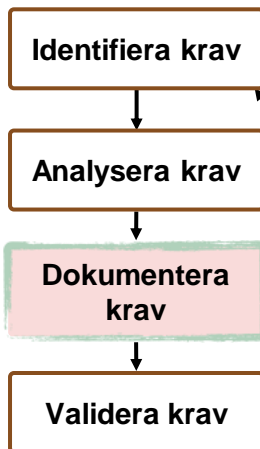


LUNDS
UNIVERSITET

F1

Intro, gruppindelning, kravhantering

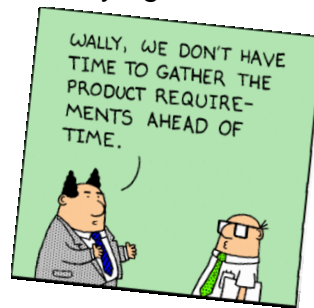
Grundbult i lyckade mjukvaruprojekt!



- Funktionella krav
- Kvalitetskrav

- Korrekt, komplett, otvetydigt, verifierbart etc.

- Skall-krav
- Användningsfall



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

Ö1a

Kravhantering

- Kursregistreringssystem
 - Intressentanalys
 - ER-diagram
- Cykeldatorprojekt
 - Kravidentifiering
 - Kravanalys
- Läskautomat
 - Interaktionskrav

R2-3: Quality Criteria for Functional and Quality Requirements

Good requirements should be:

- Correct
- Complete
- Unambiguous

R6: Add maintenance to the can machine

Interaction requirements describe the interactions between two functions of the system = combinations.

It often also collects all "strange" cases (e.g. unlikely exception scenarios)

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

Ö1b

Krav-workshop kring användningsfall 1

Användarfall:

Ta in cykel i garaget – komplett?

Korrekt	
Heltäckande	?
Otvetydigt	✓
Konsistent	?
Verifierbar	✓
	✓
	¼

Användarfall 1: Cykelägare lämnar in
 Huvudaktör: Cykelägare
 Förhandsvillkor: Cykelns streckkod finns registrerad som inlämnad

Framgångsscenario:

1. Cykelägaren läser cykelns streckkod vid ingången.
2. Ingångsdörrens lås öppnas och inlämnad.
3. Cykelägaren placerar sin cykel i genom extrautgången.

Tre MYCKET viktiga frågor:

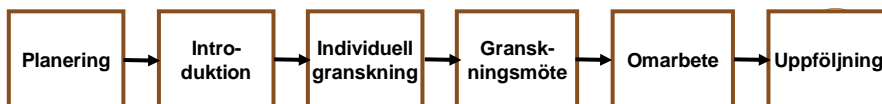
1. Är det VERKLIGEN så beställaren vi ha det?
 → Affärsmål
 → Produktmål
2. Finns det bättre lösningar?
3. Finns det undantag eller varianter?
 - Användarmissstag?
 - Säkerhet?
 - Tekniska problem?



LUND
UNIVERSITET

F2

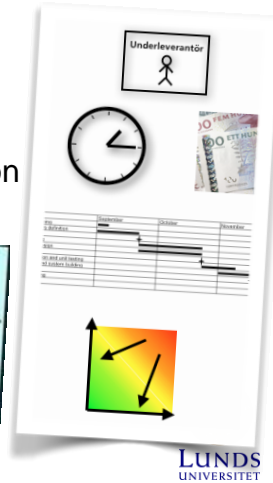
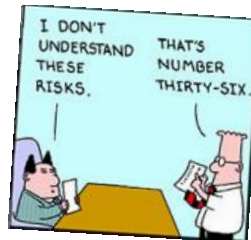
Projektplanering, granskningsprocess



Ö2

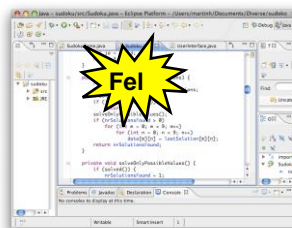
Projektplanering, dokumentgranskning

- Cykeldatorprojekt
 - Riskidentifiering
 - Riskhantering
- Projektplanering
- Granskning av aktuell kravspecifikation

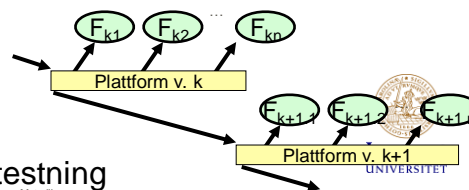


Lund University | Computer Science | Markus Borg | ETSA01 Ingenjörprocessen - Metodik

F3 Testning (blackbox), konfig.hantering, produktlinjer



- Verifiering & validering
- Olika nivåer
 - Enhetstest
 - Integrationstest
 - Systemtest
 - Acceptanstest
- Ekvivalenspart. / parvis testning



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

Ö3

Testplanering, systemtest

T4: What's the problem with the coverage matrix

Req	Test case													
	1	2	3	4	5	6	7	8	9	10	11	12	13	
2.5.1.1														
2.5.2.1														
2.5.2.2														
2.5.2.3														
2.5.3.1														
2.5.3.2														
2.5.4.2														
2.5.5.1														
2.5.5.2														
2.5.6.1														
2.5.7.3														

T5: Scope of TC6 # 2-6

Test case 6: Timeout

Pre-condition: Machine in start state.

Post-condition: Machine in start state. A water can is received.

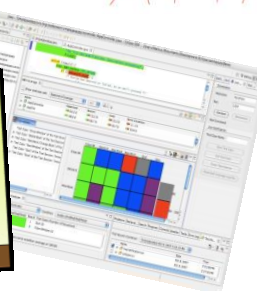
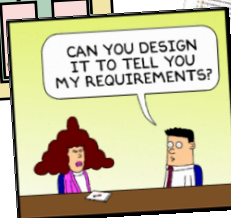
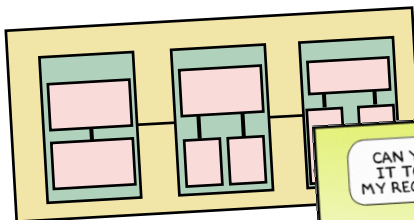
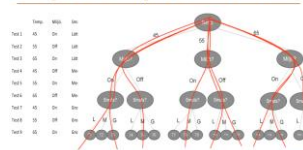
1. Press the beer selection button.
2. Wait 2 minutes.
3. Press the water selection button.
4. Insert a 5kr coin.
5. Insert 2 x1kr.
6. Receive a water can.



F4 Design, whitebox testing och verktygsdemo

- Designarkitektur beror på kvalitetskrav
- Objektorienterad design i UML
- Design av användargränssnitt
- Whitebox och kodtäckning

Exempel: Parameterpar i en diskmaskin



Ö4a Kodgranskning, teststrategi

T9: Defect log for program
Calculate statistics

nr	line	description	Type	Inject	Remove

T11: Why planned coverage <100%?



When to stop?

Name	Statement	Branch	Loop	Term	?-Operator	Synchronized
ETS401_Coverage	83,0 %	73,8 %	?	73,1 %	?	?
IDTest	83,0 %	73,8 %	?	73,1 %	?	?
IDNumber	83,0 %	73,8 %	?	73,1 %	?	?
IDNumber	100,0 %	-	?	-	?	?
assignUserCategory	80,0 %	71,4 %	?	66,7 %	?	?
calculateUserCategory	55,6 %	64,3 %	?	62,5 %	?	?
checkDateFormat	62,5 %	64,3 %	?	57,1 %	?	?
checkDigits	100,0 %	100,0 %	?	100,0 %	?	?
checkDigits	100,0 %	100,0 %	?	100,0 %	?	?
getUserCategory	100,0 %	-	?	-	?	?
registerDateOfBirth	90,0 %	83,3 %	?	83,3 %	?	?
verifyChecksum	90,9 %	87,5 %	?	91,7 %	?	?

This is probably too low!



Ö4b Test-workshop kring användningsfall 1

Användarfall:
Ta in cykel i garaget

- Vilka testfall?
- Vilka detaljer
- Vad behöver fungera först?
- Vilken ordning?
- Hur rapporterar vi resultatet (förklara i testplanen)

Framgångsscenario:

1. Cykelägaren läser cykelns streckkod m h a streckkodsläsare vid ingången.
2. Ingångsdörrens lås öppnas och cykeln registreras som inlämnad.
3. Cykelägaren placerar sin cykel i garaget och lämnar garaget genom extrautgången.

Undantag:

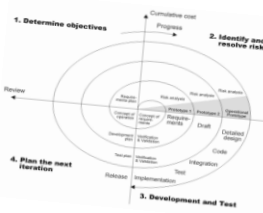
- 1a. Streckkoden finns inte i systemet
 - * PIN-kodsterminalens LED blickar rött i 2 sekunder
 - * Systemet registrerar den okända koden som ett inträngsförsök
 - * Dörren öppnas INTE
- 3a. Cykeln lämnas aldrig i garage
 - * Cykeln registreras som inlämnad (felaktigt)



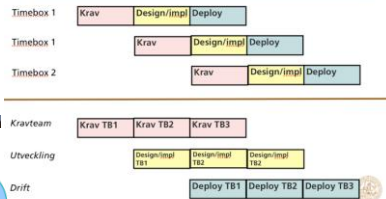
F5 Utvecklingsprocesser, vidareutveckling



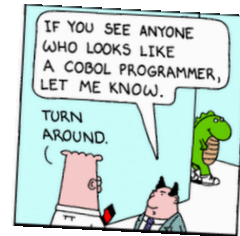
- Processmodeller
- Linjära
 - Evolutionära
 - Iterativa
 - Lätrörliga



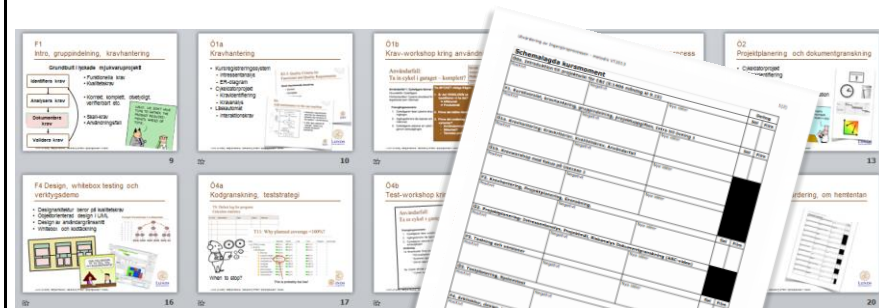
Timeboxing



Legacy systems



F6 Snabbpris, utvärdering, om hemtentan





VÄND!



Snabbspolande utvärdering av kursen

Exempel sida 2:

Cykelgarageprojektet

Del-
trog Läro-
rikt 1-
10

Kravhantering			X	8
Positivt	Negativt Svårt logga in på Google-kontot	Nya idéer Fler konkreta exempel		



Kravhantering

Produktmål

Kontextdiagram

Användningsfall

Funktionella krav

Kvalitetskrav



LUNDS
UNIVERSITET

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

Kravspecifikation: Återkoppling

#	Pos.	Checklist*	Problem	Importance
SRS1	1	1	Saknar referens till projektplanen i ex från avsnitt 2 eller 2.1	b
SRS2	2	1	borde inte beaktaren kunna förstå kravspecifikationen också?	b
SRS3	3	5	a user database Som users använder eller innehåller users?	b
SRS4	3		Kontext: diagram saknas i dokumentet. Se: http://forvalds.ca/fin/au/ETSA01/problem	a
SRS5	3		Kontext: diagram från tidigare versionen en del sällan till och från hårdvara stämmer inte med de som givits i utgåffbeskrivningen. Det är viktigt att de stämmer exakt - annars får ni problem med test av RA.1.1. Signaler till och från operatörsgränssnittet styr ni själva över.	a
SRS6	4	5	Rubriken otydlig blandning av Interface och User interface	b
SRS7	RA.1.2	5.1	Mycket på en gång. Vad som menas med Display information är otydligt.	a
SRS8	RA.1.3	10	Vad menas med sample statistics. Specificera här eller i 4.2.1.8	a
SRS9	RA.1.4	3	Låset tar inga lock-kommandon	a
SRS10	RA.1.8	3	PIN-code terminal skickar tecken för tecken	a
SRS11	RA.2.1.8	4	Ej tillräckligt specificerat för att kunna testas (se SRS7 ovan)	a
SRS12	RA.2.1.8	3	Vilka konsekvenser får detta om värdet sätts till litet än antal parkerade cyklar	b
SRS13	RA.2.1.9	5	föreslår ej in this category. Menar ni sort on check in-time?	b
SRS14	4.2.2	9	Detta känns väldigt mycket som designförning och kommer att tolkas en hel del white-box-testing. Forsök skriva om så att endast det som berör användaren (beteenden och systemets reaktioner) berörs.	a
SRS15	4.2.3	9	Se ovan	a
SRS16	RA.2.3.5	3	Betyder detta att om man står fel kod måste vänta 5 sekunder på att få göra ett nytt försök? Hur vet användaren att när man får göra ett nytt försök?	b
SRS17	RA.3.2-3	11	Liknar varandra mycket	a
			Data måste kunna överleva en system crash eller uppstart och	a



LUNDS
UNIVERSITET

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

Projektplanering

Tidsestimat

Aktiviteter

Leverabler

Gantt-schema

Risikanalys



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

Projektplan: Återkoppling

#	Pos.	Checklist*	Problem	Importance
SRS1	1	1	Saknar referens till projektplanen i ex från avsnitt 2 eller 2.1	b
SRS2	2	1	berörs inte besöffaren kunna förstå kravspecifikationen också?	b
SRS3	3	5	a user database Som users använder eller innehåller users?	b
SRS4	3		Kontext: diagram saknas i dokumentet. Se: http://forvalds.ca/lin/au/ETSA01/problem	a
SRS5	3		Kontext: diagram från tidigare versionen av del sänder till och från hårdvara stämmer inte med de som givits i uppgiftsbeskrivningen. Det är viktigt att de stämmer exakt - annars får ni problem med test av RA.1.1. Signaler till och från operatörsgränssnittet styr ni själva över.	a
SRS6	4	5	Rubriken otydlig blandning av Interface och User interface	b
SRS7	RA.1.2	5.1	Mycket på en gång. Vad som menas med Display information är otydligt.	a
SRS8	R.4.1.3	10	Vad menas med sample statistics. Specificera här eller i 4.2.1.8	a
SRS9	R.4.1.4	3	Låst tar inga lock-kommandon	a
SRS10	R.4.1.8	3	PIN-code terminal skickar tecken för tecken	a
SRS11	RA.2.1.6	4	Ej tillräckligt specificerat för att kunna testas (se SRS7 ovan)	a
SRS12	RA.2.1.8	3	Vilka konsekvenser får detta om värdet sätts till litet än antal parkerade cyklar	b
SRS13	RA.2.1.9	5	föreslår ej in this category. Menar ni sort on check-in-time?	b
SRS14	4.2.2	9	Detta känns väldigt mycket som designplanering och kommer att betyda en hel del white-box-testing. Forsök skriva om så att endast det som berör användaren (betendens och systemets reaktioner) berörs.	a
SRS15	4.2.3	9	Se ovan	a
SRS16	RA.2.3.5	3	Betyder detta att om man slår fel kod måste vänta 5 sekunder på att få göra ett nytt försök? Hur vet användaren att när man får göra ett nytt försök?	b
SRS17	RA.3.2-3	11	Liknar varandra mycket	a
			Data måste kunna överleva en system crash eller uppstart och	



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

Testplanering

Testprocess

- Mål och metoder
- Testrapporter

Kravtäckning

Testfall

- Pre- & post-condition
- Observerbart
- Genomförbart



Testplan: Extern QA-granskning

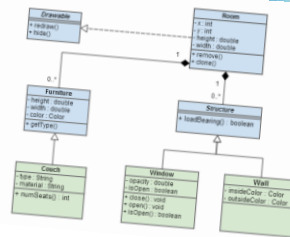
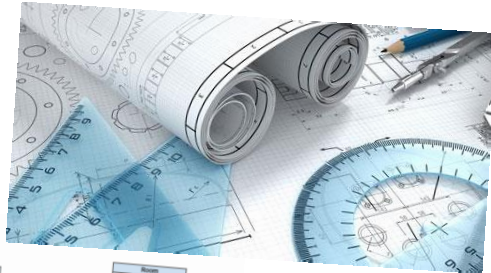


Design

Klassdiagram med relationer

Klassbeskrivningar:

- Publika metoder
- Attribut
- Parametrar



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



Designdokument: Återkoppling

#	Pos.	Checklist*	Problem	Importance
SRS1	1	1	Saknar referens till projektpåren i ex från avsnitt 2 eller 2.1	b
SRS2	2	1	Börde inte beaktas kunna förstå kravspecifikationen också?	b
SRS3	3	5	a user database som users använder eller innehåller users?	b
SRS4	3		Kontext: diagram saknas i dokumentet. Se: http://forvalds.ca/lin/soE/TSAD/1/problem	a
SRS5	3		Kontext: diagram från tidigare version: en del sällan till och från hårdvara stämmer inte med de som givits i uppgiftsbeskrivningen. Det är viktigt att de stämmer exakt - annars får ni problem med test av RA.1.1. Signaler till och från operatörsgränssnittet styr ni själva över.	a
SRS6	4	5	Rubriken otydlig blandning av Interface and User interface	b
SRS7	RA.1.2	5.1	Mycket på en gång. Vad som menas med Display information är otydligt.	a
SRS8	R.4.1.3	10	Vad menas med sample statistics. Specificera här eller i 4.2.1.8	a
SRS9	R.4.1.4	3	Låst tar inga lock-kommandon	a
SRS10	R.4.1.8	3	PIN-code terminal skickar tecken för tecken	a
SRS11	RA.2.1.6	4	Ej tillräckligt specificerat för att kunna testas (se SRS7 ovan)	a
SRS12	RA.2.1.8	3	Vilka konsekvenser får detta om värdet sätts till litet än antal parkerade cyklar	b
SRS13	RA.2.1.9	5	försäkr ej in this category. Menar ni sort on check-in-time?	b
SRS14	4.2.2	9	Detta känns väldigt mycket som designförslag och kommer att betyda en hel del white-box-testing. Förskä skrivs om så att endast det som berör användaren (beteenden och systemets reaktioner) berörs.	a
SRS15	4.2.3	9	Se ovan	a
SRS16	RA.2.3.5	3	Betyder detta att om man skär fel kod måste vänta 5 sekunder på att få göra ett nytt försök? Hur vet användaren att när man får göra ett nytt försök?	b
SRS17	RA.3.2-3	11	Liknar varandra mycket	a
			Data måste kunna överleva en system crash eller uppstart och	



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



Dokumentgranskning

Kända fel i kravspecifikation till "ABC Video System"

Position	Checklistepunkt	Kommentar
Kap. 1.3	6	Termen "rentable" definierad men används ej i resten av dokumentet. Istället används andra ord, "tape", "video", etc. Start om detta är samma sak.
Kap. 1.3	6	Definierade termer: "rental", "transaction"
Kap. 1.4	6	Referens -> Chapter
Kap. 2	2	Formulering så mycket krävs som möjligt. Leder lätt till missförstånd.
Text innan kap. 2.1	6	"due amount", "due amount" för svårt att förstå.
Kap. 2.1	2	Okänt system ska köras på en eller flera PC.
Kap. 2.4	1	Punkt 3.1.9 under "Output" hör förmodligen inte hemma i kravspecifikationen för videosystemet. Dessutom är de delvis redundanta.
Kap. 3.1	6	Ej fullständig beskrivning av formatet. Varför inget starttillstånd?
FR 1	2	"Daily reports" med "password" saknas jämfört med kap. 2.3.
FR 1	7	Det borde framgå vilken kravsektion i dokumentet som man hör till vilket menval.



LUNDS
UNIVERSITET

Programmering

```

class VideoSystem {
public:
    VideoSystem() {}
    VideoSystem(int capacity) : capacity(capacity) {}
    VideoSystem(int capacity, string name) : capacity(capacity), name(name) {}
    VideoSystem(int capacity, string name, string password) : capacity(capacity), name(name), password(password) {}

    void addVideo(string title, int duration) {
        if (duration > 0) {
            videos.push_back({title, duration});
        }
    }

    void listVideos() const {
        for (const auto& video : videos) {
            cout << video.title << " (" << video.duration << " min)" << endl;
        }
    }

    void searchVideos(const string& keyword) const {
        for (const auto& video : videos) {
            if (video.title.find(keyword) != string::npos) {
                cout << video.title << " (" << video.duration << " min)" << endl;
            }
        }
    }

    void listReports() const {
        for (const auto& report : reports) {
            cout << report.title << " (" << report.duration << " min)" << endl;
        }
    }

    void searchReports(const string& keyword) const {
        for (const auto& report : reports) {
            if (report.title.find(keyword) != string::npos) {
                cout << report.title << " (" << report.duration << " min)" << endl;
            }
        }
    }

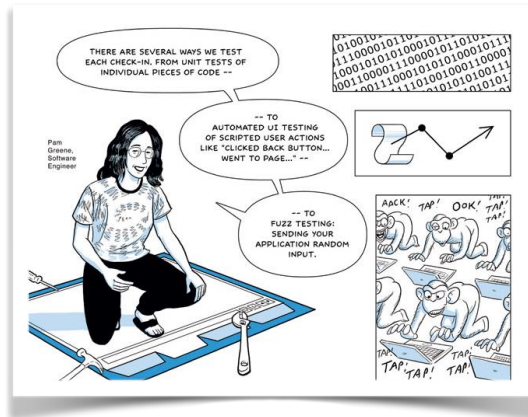
private:
    vector<Video> videos;
    vector<Report> reports;
    int capacity;
    string name;
    string password;
};

```



LUNDS
UNIVERSITET

Testning



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



Processförbättring för kursen

Kursen kommer att göras om: utmärkt läge för feedback!

Försvinner från D1 → ersätts med annan kurs

Kvarstår som obligatorisk för C1 → utökas med 1 HP

Fortsatt alternativobligatorisk för I3

Konsekvenser

- Färre än hälften så många studenter → minskad projektadministration
- En extra poäng att fylla med innehåll

SVARA PÅ CEQ-ENKÄTEN – Den har chans att påverka mycket!



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

www.lth.se/stress-tarmflora

Killar 19-35 år

Web-formulär (2 min)

Lågstressade och Högstressade tar prover hemma, lämnar på Kem centrum eller BMC

stress-tarmflora@food.lth.se, 046-2228326



LUNDS
UNIVERSITET

Om hemtentamen

ETSA01 Ingenjörprocessen 1 - Metodik VT15 | Markus Borg



Allmänt om hemtentamen

Uppgiften görs tillgänglig 2 juni kl 8:00, lämnas in senast 3 juni kl 10:00

Uppgifterna löses individuellt

Tanken att det ska ta ungefär en arbetsdag att svara på frågorna

Allt material i kursen ingår. Alla svar ska baseras på kursboken eller annat material i kursen. Man kan referera till ytterligare material, då ska alla källor tydligt redovisas.

Svar på olika nivåer:

- Godkänd nivå: en grundläggande förståelse av innehållet, grundläggande kunskap om vad i materialet som är viktigt, återge fakta på ett korrekt sätt.
- Mer avancerad nivå: analysera materialet genom t ex jämförelser och identifiering av mönster, göra jämförelser med projektet i kursen.



LUNDS
UNIVERSITET

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

Betyg i kursen

Kursbetyget är en sammanräkning av projektbetyg och tentaresultat

Skriver ni en riktigt bra hemtentamen så får ni betyg 5 – Bara projektet är ok!

Har er grupp gjort ett riktigt bra projekt så sänks betygsgränserna på tentan

Riktlinje:

Projektbetyg	Tentamen	Slutbetyg
	>30, <44	3
3	>44, <54	4
	>54	5
	>30, <40	3
4	>40, <50	4
	>50	5
	>30, <36	3
5	>36, <46	4
	>46	5



LUNDS
UNIVERSITET

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

Språket viktigt i hemtentamen!

Förväntningar på språket högre än vid salstenta

- Teknisk dokumentation – undvik talspråk
- Använd rättstavningsprogram
- Särskriv inte
- Använd rätt terminologi
- Läs genom innan inlämning. Två gånger!

Skriv på svenska eller engelska

Ni får gärna rita figurer/tabeller etc.



Praktiskt om hemtentamen

Dokument och omfattning:

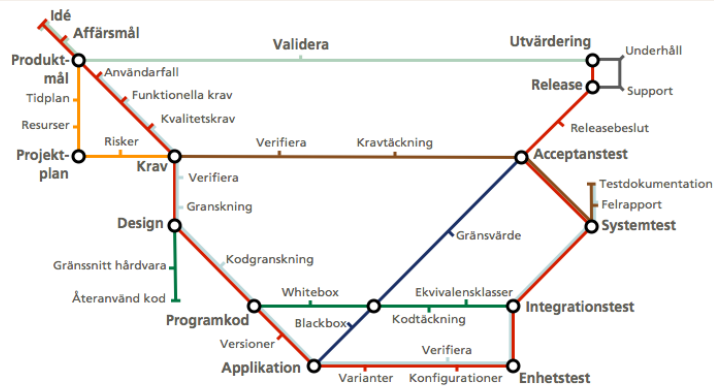
- Rapport på 4-6 sidor text - helst inte mer
 - Irrelevanta inslag ger avdrag
 - Kvalitet snarare än kvantitet!

Inlämning x 3:

- För rättning:
 - etsa01@cs.lth.se – dvs. samma som för projektets slutinlämning
 - Ärende "ETSA01 Tentamen <samID>"
 - SamID meddelas per mail
- För plagiatkontroll:
 - Mail skickas med CC: markus.borg.lu@analys.urkund.se
- För autentisering och säkerhet
 - Pappersutskrift i grå skåpet eller per papperspost
 - Underskrift!



OM det hade varit en salstenta: Förklara begreppen och hur saker och ting hänger ihop



Hemtenta innebär mer resonande uppgifter utan tydligt rätt eller fel. Ni ska kunna drive ett resonemang och argumentera för era svar med rätt begrepp.



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

Tentafråga VT 2012

U1 Test hela vägen (15 p)

Beskriv de olika nivåerna på dynamisk testning under ett mjukvaruutvecklingsprojekt från enhetstestning till acceptanstest. För varje nivå bör beskrivningen åtminstone svara på följande frågor: Vem? Vad? När? Hur? Hur länge? Varför? Använd åtminstone följande begrepp i rätt sammanhang:

Acceptanstest

Black-box-test

Branch coverage

Enhetstest

Gränsvärdestestning

Integrationstest

Kravtäckningsmatris

Parvis testning

Regressionstest

Stopkriterium

Stresstest

Systemtest

Testprotokoll

Validering

Verifiering

White-box-test



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

Tentafråga VT 2012

Förslag

U1 Test hela vägen (15 p)

- Förstå skillnad på **verifiering/validering** samt **statisk/dynamisk** testning

Enhetstestning

- Av utvecklarna medan källkod skrivs, whitebox på klassnivå
- Testsvit används som regressionstest

Integrationstest

- Av utvecklarna efter hand som klasser blir klara, whitebox med kodtäckning, stoppkriterium?

Systemtest

- Av testarna, testar helheten, parvis testning, gränsvärden
- blackbox baserat på kraven mot kravtäckningsmatris
- Stresstest?

Acceptanstest

- Av testarna med kundrepresentant, blackbox



Tentafråga VT 2012



U2 Hög standard på programkod (10 p)

I Jalote avsnitt 7.1.4 presenteras kodningsstandarder för fyra olika områden.

- Diskutera fördelar och nackdelar med att använda kodningsstandarder av dessa slag och om du eventuellt kan se någon skillnad vad gäller fördelar/nackdelar för de fyra områdena.
- Många gånger kan de vara svårt för en organisation att hålla sig till en gemensam kodningsstandard i den utsträckning som Jalote presenterar. Diskutera varför det kan vara så.
- Anta att du fick i uppdrag att verka för att en organisation för programvaruutveckling skulle följa de standarder som Jalote presenterar. Föreslå och argumentera för hur du skulle gå tillväga för att följa upp om standarderna efterlevs.

File	Line	Column	Severity	Message
src/main/java/com/example/HelloWorld.java	1	1	Warning	File name should not contain spaces
src/main/java/com/example/HelloWorld.java	1	1	Warning	File name should not contain spaces
src/main/java/com/example/HelloWorld.java	1	1	Warning	File name should not contain spaces
src/main/java/com/example/HelloWorld.java	1	1	Warning	File name should not contain spaces
src/main/java/com/example/HelloWorld.java	1	1	Warning	File name should not contain spaces
src/main/java/com/example/HelloWorld.java	1	1	Warning	File name should not contain spaces
src/main/java/com/example/HelloWorld.java	1	1	Warning	File name should not contain spaces
src/main/java/com/example/HelloWorld.java	1	1	Warning	File name should not contain spaces
src/main/java/com/example/HelloWorld.java	1	1	Warning	File name should not contain spaces
src/main/java/com/example/HelloWorld.java	1	1	Warning	File name should not contain spaces



Tentafråga VT 2012

Förslag

U2 Hög standard på programkod (10 p)

Naming conventions, files, statements, documentation comments

a) För och nackdelar

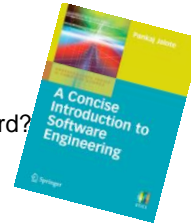
- + läsbarhet, enklare buggrättningar, förenklat underhållsarbete
- finns alltid undantag där det passar sämre, konventioner kan innebära merarbete

b) Varför svårt för en hel organisation att använda samma standard?

- mycket är en smaksak, svårt att tvinga
- organisationer kan vara heterogena

c) Hur följa upp att standarder efterlevs?

- utse en kodkvalitet-ansvarig som bjuds in till alla granskningsmöten
- introducera verktyg för att kontrollera att konventionerna följs



LUNDS
UNIVERSITET

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

Tentafråga VT 2012



U3 Samma sak på olika sätt (15p)

Ett *användarfall* (use-case), ett *acceptanstestfall* och ett *scenario* i en *användarmanual* kan mycket väl beskriva i princip samma tänkta händelseförlopp.

- a) Börja med att formulera ett eget *användarfall* på kanske 5-10 punkter för vilket system som helst som **inte** har att göra med cykelgarageprojektet eller de andra exemplen som använts i kursen. Systemet kan finnas på riktigt eller vara påhittat.

Formulera därefter ett (1) *acceptanstestfall* och en (1) *manualltext* som relaterar till användarfallet.

Inled med en kort beskrivning av ditt tänkta system så man förstår användarfallets sammanhang. Presentera sedan gärna användarfallet, testfallet och manualltexten i en tabell med tre kolumner så att de blir lättare att jämföra med varandra.

- b) Jämför de tre olika beskrivningarna och diskutera varför de skiljer sig åt.



LUNDS
UNIVERSITET

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

Tentafråga VT 2012

Förslag

U3 Samma sak på olika sätt (15p)

System för att administrera **badmintontävlingar**. Exempel: **registrera match**

Användningsfall

1. Skanna QR-kod på rapport
2. Mata in set-siffror
3. Mata in start-/sluttid
4. Mata in antal använda bollar
5. Klicka på 'Bekräfta'

Testfall

1. Skanna QR-kod på rapport TEST_A
2. Mata in set-siffrorna "15-21, 21-17, 21-8"
3. Mata in starttid "10:25" och sluttid "11:30"
4. Mata in antal använda bollar: 12
5. Klicka på 'Bekräfta'
6. Verifiera att "Match 22" presenteras i lista 2

Scenario i manual

1. Använd QR skannern för att registrera din match
2. Fyll i matchens resultat, uppdelat per set
3. Fyll i klockslag då matchen började resp. slutade
4. Mata in hur många fjäderbollar som användes
5. Klicka på 'Bekräfta' för att lagra matchresultatet



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



LUND
UNIVERSITET

Tentafråga VT 2012



U4 Spårbarhet (10p)

Raja and Kamran genomförde 2008 en systematisk kartlägningsstudie som en del av ett examensarbete i software engineering vid Blekinge Tekniska Högskola. En av författarnas forskningsfrågor rörde vilken definition av spårbarhet som det oftast refereras till i vetenskapliga artiklar. Studenterna fann att den definition som vanligast använts är:

"(Requirements) traceability refers to the ability to describe and follow the life of a requirement, in both a forward and backward direction (i.e. from its origin, through its development and specification, to its subsequent deployment and use, and through periods of on-going refinement and iteration in any of these phases)" (Gotel and Finkelstein, 1994

Frågor:

- a) I cykelgarageprojektet underhölls spårbarhet mellan systemkrav och systemtester. Ge exempel på aktiviteter i ett programvaruutvecklingsprojekt som kan underlättas av att denna typ av spårbarhet finns tillgänglig.
- b) Spårbarhet kan även existera mellan andra typer av utvecklingsartefakter, t.ex. kravkällkod, källkod-testfall, källkod-manual och defektrapport-krav. Att bestämma vilken typ av spårbarhet som ska underhållas innebär ett avvägande. Redogör för nackdelar med en hög nivå av spårbarhet. Föreslå även exempel på typer av projekt där spårbarhet trots detta brukar anses nödvändigt.

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

Tentafråga VT 2012

Förslag

U4 Spårbarhet (10p)

Underlättas av spårbarhet:

- Buggrättning
- Ändringshantering
- Vidareutveckling
- Regressionstestning
- Generell systemförståelse
- ...



Nackdelar med spårbarhet. När nödvändigt?

- dyrt att upprätta, särskilt i legacy-system
- krävs mycket arbete att underhålla vid förändringar, svårt för agil utveckling
- lätt att peka ut fel kopplingar, t.ex. för många
- ingår i standarder för säkerhetskritisk utveckling



Tentafråga VT 2012

U5 Statisk testning (10 p)

Beskriv hur processen för dokumentgranskning bör gå till. Ange t. ex. inblandade roller, delaktiviteter och delresultat. Diskutera kostnader i form av arbetstid och ledtid, d.v.s. hur lång tid man kan tänkas behöva mellan de olika aktiviteterna för att det ska vara realistiskt att genomföra hela granskningsprocessen.

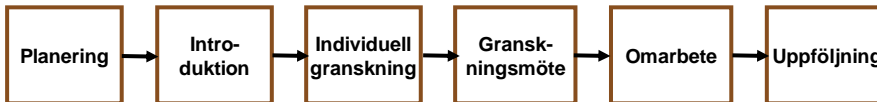
Ge exempel på typer av fel och problem som är lättare att identifiera vid dokumentgranskning än vid dynamisk testning. Motivera dina svar.



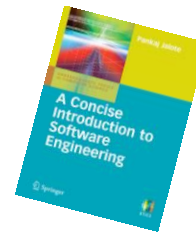
Tentafråga VT 2012

Förslag

U5 Statisk testning (10 p)



- Roller vid mötet: ordförande, sekreterare, författare, granskare
- Rimligt med inbjudan 1-2 veckor innan mötet
- Individuell granskning: 5-15 sidor/h beroende på komplexitet
- Granskningsmöte: max 2 h
- Uppföljningsmöte en vecka senare



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

Fler exempel på tentafrågor

Om | Utbildning | Forskning | Nyheter | Kalendarium | Kontakt | Intern

ETSA01 - Ingenjörprocessen, metodik > Tentamen

Nyhetsarkiv

Kursprogram

Projekt 2015

Föreläsningar

Övningar

Tentamen

- o Till SAM-Id

Tentamen

Omtenta 2015-01-08 -- 09

Omtenta: 2015-01-08 kl 08.00 till 2015-01-09 kl 10.00 OBS! Obligatorisk föranmälan till Jonas.Wisbrant@cs.lth.se senast den 2015-01-05.

Hemtentamen skickas sedan via mail kl 2015-01-08 kl 08.00. Hör av dig om du fortfarande inte fått den kl 08.15. Jonas Wisbrant 0730-24 96 75

Fördyliganden kan komma att publiceras på denna sida under dagen.

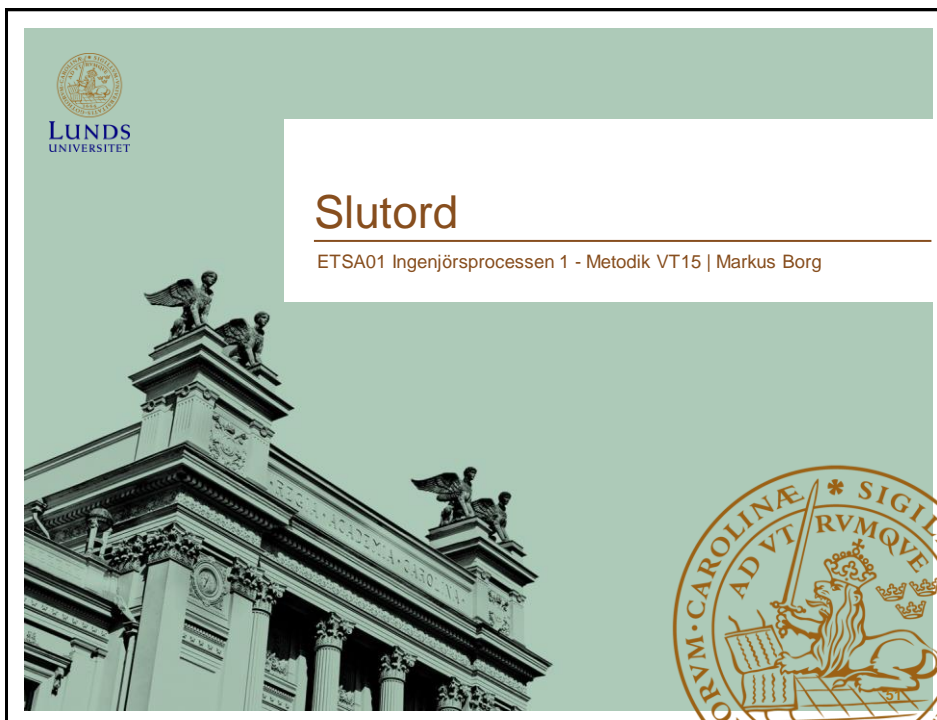
Jonas Wisbrant finns i princip tillgänglig per telefon (046-222 34 83 och/eller 0730-24 96 75) för att t ex förtydliga frågor under perioderna

- 8/1 kl 09.30 - 11.00 och 14.00 - 20.00
- 9/1 kl 08.00 - 09.30

GAMLA TENTOR

- 2014-06-02
- 2013-05-30
- 2012-05-24
- 2011-05-26
- 2010-06-02
- 2009-08-21
- 2009-06-01

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



Slutord

ETSA01 Ingenjörprocessen 1 - Metodik VT15 | Markus Borg

Kursmål - Beställningen i text



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**

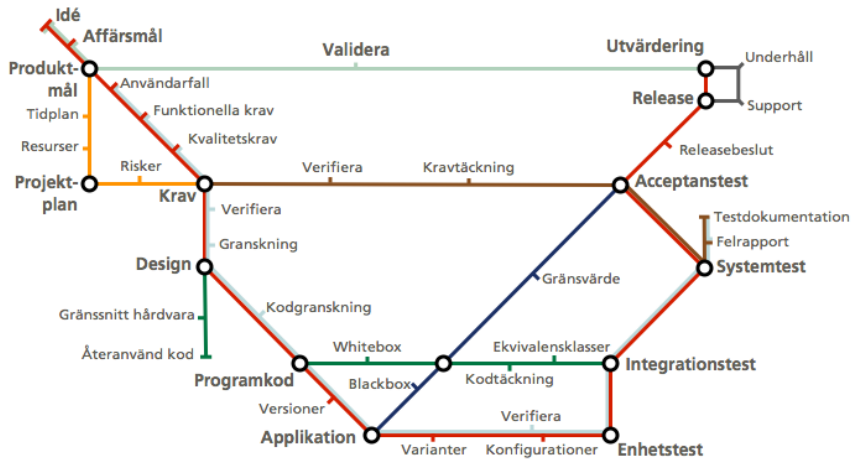
Färdighet och förmåga

- 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**

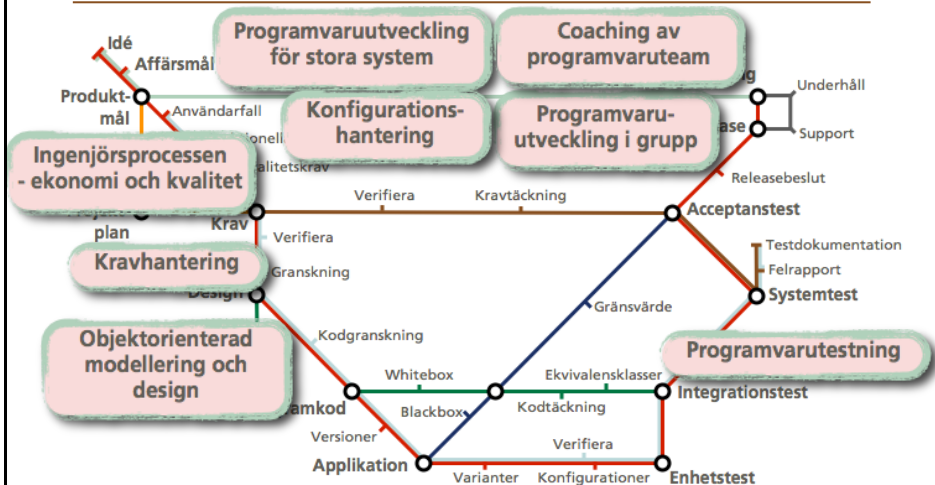
Värderingsförmåga och förhållningssätt

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

En sista överblick



Gå djupare!



Frågan kvarstår

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

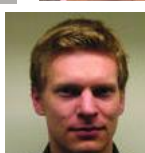
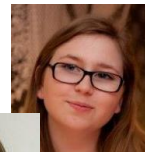
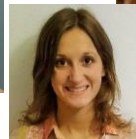


Kan man förstå vad som händer i stora programvaruprojekt utan att ha studerat software engineering?



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

Vi tackar för oss...



... och hoppas att ni i och med den här kursen har fått fler frågor än svar – men en grund för vidare funderingar!



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