



LUND UNIVERSITY

ETS170 Kravhantering

Föreläsning 4:

Specificering del 2: Lau:3.6-3.16, 4; Wei; Ka1:3.2

Riktlinjer för tentaprobleminlämningsuppgifter

Björn Regnell
<http://www.cs.lth.se/ETS170>

Kriterier för projektbetyg (se hemsidan)				
3=Acceptable 4=Good 5=Excellent				
Project Deliverables	Weight	Requirements for 3	Requirements for 4	Requirements for 5
PP Project Plan	0.1	Includes almost all PP items in project description in v3 and is in part up to date throughout the project	Includes all PP items in project description in v2 and document is reasonably up to date throughout the project	Includes all PP items in project description in v1 and document is completely up to date throughout the project
RD Requirements Document	0.4	RD v3 fulfills most of the RD items in the project descriptions fairly well. There are improvements between versions.	RD v2 fulfills all of the RD items in the project descriptions well, demonstrating good knowledge in how to use different techniques. There are many improvements between versions.	RD v2 is excellent and comprehensive, while significantly exceeding the expectations in the RD items of the project description, demonstrating deep knowledge in how to use many different techniques. There are radical improvements between versions.
*Context diagram *Tasks use cases scenarios *Data req (E/R VW DD) *FR+NFR				
RR Review Report	0.1	RR includes a few helpful suggestions for improvements	RR includes helpful suggestions for improvements that are all ranked for critically based on given checklist	RR includes a significant number of helpful suggestions for improvements that are all ranked for critically, based on the given checklist, but also stretching beyond the given checklist.
CP Conference Presentation	0.1	CP includes a fair description of limited results and some experiences	CP includes a good description of results and experiences, prioritized to keep the time limits and still convey the message	CP includes an excellently structured and engaging description of results and experiences with reflections, prioritized to keep the time limits and still convey the message.
EP Experience Report	0.3	EP includes a fair description of results and learning outcomes.	EP provides a good description of results and learning outcomes, including answering all questions in the EP items of the project description and an account of technique selection rationale	EP provides an excellent description of results and learning outcomes, including answering all questions in the EP items of the project description and a thorough account of technique selection rationale. Learning outcomes are related to course learning objectives in a qualified discussion.
General prerequisites		Delivered in time (late submission may be acceptable if negotiated in advance with good reasons). No single group member is free-riding on the rest of the group.	Delivered in time (late submission may be acceptable if negotiated in advance with good reasons). No single group member is free-riding on the rest of the group. Equal division of labour among group members	Delivered in time. No single group member is free-riding on the rest of the group. Equal division of labour among group members with documented effort outcomes including planned versus spent
Sum of weights	1			

Vadå "fullständig kravspec"?

- Det är i praktiken omöjligt att specificera allt i minsta detalj!
- Vad är bra nog?
-> Beror på situationen
- Tips: Fokusera på de krav som innebär störst risk för...
 - ♦ Missförstånd bland intressenter
 - ♦ Att slutresultatet ej blir önskvärt
- Spennera inte en massa tid på krav som alla känner väl till
- Konceptstudier, förstudier, möjlighetsstudier etc. för att...
 - ♦ minska risker
 - ♦ "hoppa" mellan abstraktionsnivåer



Helikoptervy över tekniker & stilar för funktionella krav

Datakravstilar:

- ✓ Datamodell (=E/R-diagr.)
- ✓ Dataordlista
- ✓ Reguljära uttryck
- ✓ Virtuella fönster

Funktionella kravstilar:

- ✓ Kontextdiagram
- ✓ Händelse- & Funktionslistor
- ✓ Produkttegenskapskrav
- ✓ Skärmbilder & Prototyper
- Uppgiftsbeskrivningar
- Egenskaper från uppgifter
- Uppgifter och stöd
- (Levande) Scenarier
- Högnivåuppgifter
- Användningsfall
- Uppgifter med data
- Dataflödesdiagram
- Standardkrav
- Krav på utvecklingsprocessen

Funktionella detaljer:

- Enkla och sammansatta funktioner
- Tabeller & Beslutstabeller
- Textuella processbeskrivningar
- Tillståndsdigram
- Övergångsmatriser
- Aktivitetsdiagram
- Klassdiagram
- Samarbetsdiagram
- Sekvensdiagram

Speciella gränssnitt

- Rapporter
- Plattformskrav
- Produktintegration
- Tekniska gränssnitt



Användningsfallsmodellering

Use case modelling

Aktör (*actor*)

- en kategori av användare, roll

Användningsfall (*use case*)

- måluppfyllande användningssituation

Scenario

- en specifik realisering

Exempel:

Bankomat: "Ta ut pengar"
(stoppa in kort, knappa in kod ...)

Ordbehandling: "Kontrollera stavning"
(välj stycke, välj ordlista ...)

Bra till vadå?

© Björn Regnell

Fördelar med dynamiska modeller av vad användaren gör

Lätt att förstå för icke-tekniska intressenter

Kan koppla ihop krav på olika nivåer

Bra för att modellera funktionella krav

Hjälper till att strukturera kraven

Ger ett dynamiskt perspektiv på kraven

Stödjer spårbarhet

Bra grund för testfall

© Björn Regnell

Fallgropar

För mycket detaljer – "överspecificering"

För lite detaljer – "underspecificering"

Fragmentering av information

För tidig design

Ej enhetliga beskrivningar

- ♦ Olika form, detaljnivå, terminologi

Inkonsekvenser

- ♦ Motstridiga användningsfall

Begränsad täckning av kraven (ofullständighet)

Funktionell nedbrytning -> dålig OO design

© Björn Regnell

Begreppsförvirring: Scenario, Användningsfall, Uppgift

Scenario=

(1) En specifik realisering av ett användningsfall; en instans av ett användningsfall

(2) En rik beskrivning av en specifik händelse; en liten "novell" med massor av detaljer (kallas av Lauesen för "vivid scenario")

(3) Alla typer av exempelbaserade kravbeskrivningar, tex användningsfall å la UML, "user stories" i XP, Tasks i usability engineering etc.

(denna betydelse råder i [Weidenhaupt])

(4) Möjliga framtida händelseförlopp (denna betydelse gäller ofta inom riskhantering)

...

Dessutom finns det många olika sätt att beskriva användningsfall utöver Jacobson, tex Cockburn, Lauesen, ...

© Björn Regnell

Användningsfall och UML



Grady Booch
(Booch notation)



James Rumbaugh
(OMT object modeling technique)



Ivar Jacobson
(use cases)

De tre "amigos" på Rational (numera uppköpt av IBM)
Andra ursprung till användningsfall:

- Scenario-based RE: Hsia, Potts, Sutcliffe, Pohl, m.fl.
- Uppgiftsbeskrivningar från användbarhetsteknik (Task description from usability engineering)

© Björn Regnell

Olika aspekter av scenario-baserade krav

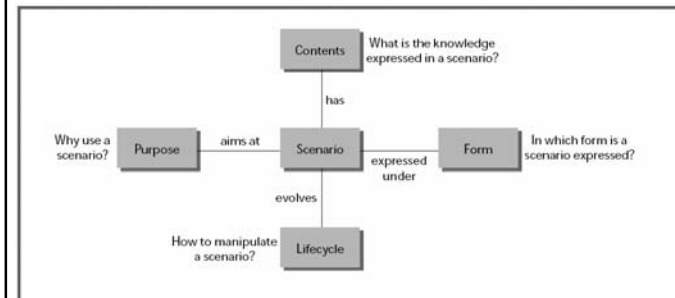


Figure 1. Four views on scenarios: form, purpose, content, and life cycle.

[Weidenhaupt et al. IEEE Software March/April 1998]

© Björn Regnell

Stor skillnad på olika scenario-baserade krav

Scenario Facet		Project										
Form	Narrative text	Form	Content	Purpose and usage	Life cycle and management	Form	Content	Purpose and usage	Life cycle and management	Form	Content	Purpose and usage
	Structured text	Form	Content	Purpose and usage	Life cycle and management	Form	Content	Purpose and usage	Life cycle and management	Form	Content	Purpose and usage
	Diagrammatic notations	Form	Content	Purpose and usage	Life cycle and management	Form	Content	Purpose and usage	Life cycle and management	Form	Content	Purpose and usage
	Images	Form	Content	Purpose and usage	Life cycle and management	Form	Content	Purpose and usage	Life cycle and management	Form	Content	Purpose and usage
	Animations or simulations	Form	Content	Purpose and usage	Life cycle and management	Form	Content	Purpose and usage	Life cycle and management	Form	Content	Purpose and usage
Typical size (pages)		Form	Content	Purpose and usage	Life cycle and management	Form	Content	Purpose and usage	Life cycle and management	Form	Content	Purpose and usage
System context		Form	Content	Purpose and usage	Life cycle and management	Form	Content	Purpose and usage	Life cycle and management	Form	Content	Purpose and usage

Vill man prata "use case" behöver man förtydliga form, innehåll och syfte i det specifika fallet

[Weidenhaupt et al. IEEE Software March/April 1998]

© Björn Regnell

Scenario Facet		Project										
		1	2	3	4	5	6	7	8	9	10	11
Form	Narrative text	F	M	F	M	M	M	F	M	—	—	M
	Structured text	F	F	M	F	F	F	F	M	—	F	F
	Diagrammatic notations	—	—	M	F	—	M	F	F	—	F	—
	Images	F	F	M	M	M	—	M	M	—	M	—
	Animations or simulations	—	—	—	—	—	—	—	—	—	—	—
Content	Typical size (pages)	1-3	1-2	2-8	3-8	10-20	5-20	7	7	1-2	3-10	10-50
	System context	M	M	M	M	—	—	F	F	—	F	—
	System interaction	F	F	F	F	F	F	F	F	—	M	F
	Internal system	—	—	M	—	—	—	M	F	F	—	—
	Concretization of abstract models	F	M	F	F	F	F	F	F	—	F	M
Purpose and usage	Scenarios instead of abstract models	—	—	F	F	—	—	—	—	—	—	—
	Interdisciplinary development	F	M	F	F	F	F	F	F	—	F	F
	Scenario use with prototypes	F	F	F	F	M	—	—	—	—	F	F
	Complexity reduction	F	F	F	F	F	M	F	M	F	F	F
	Agreement and consistency	F	F	F	F	F	F	F	F	M	F	F
Life cycle and management	Scenario use with glossaries	—	—	F	M	M	—	—	—	—	—	—
	Reflection on static models	F	F	F	F	F	F	F	—	—	F	F
	Partial views	M	M	F	F	F	M	F	F	F	M	—
	Distributed scenario development	M	—	F	F	F	M	F	F	M	—	—
	Review	F	F	F	F	F	F	M	M	M	M	F
Life cycle and management	Traceability issues	F	M	F	F	F	M	F	F	M	M	M
	Basis for test cases	M	—	M	M	M	M	—	M	M	M	M
	Evolution	F	M	F	F	F	F	M	F	F	F	F

* 80% of the screens of the end product.

© Björn Regnell

Fig 3.6A Task descriptions

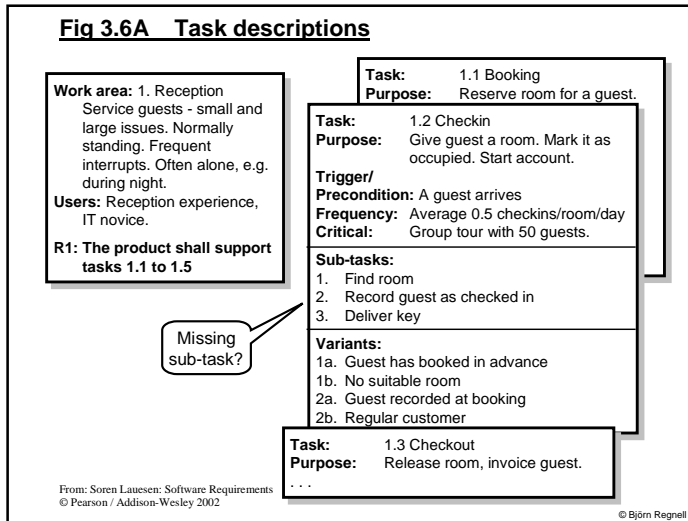


Fig 3.6B Triggers, options, preconditions

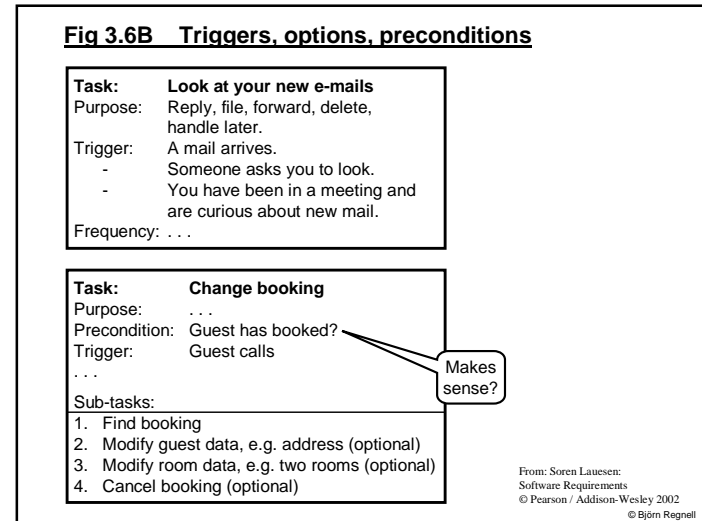


Fig 3.8A Tasks & Support

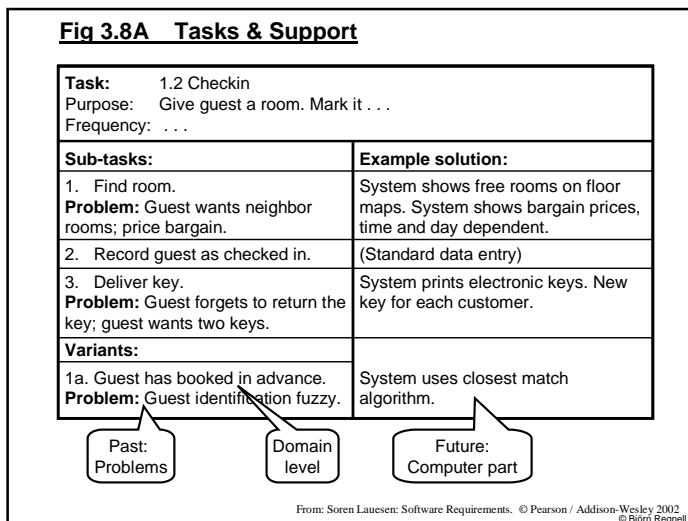


Fig 3.9 Vivid scenario

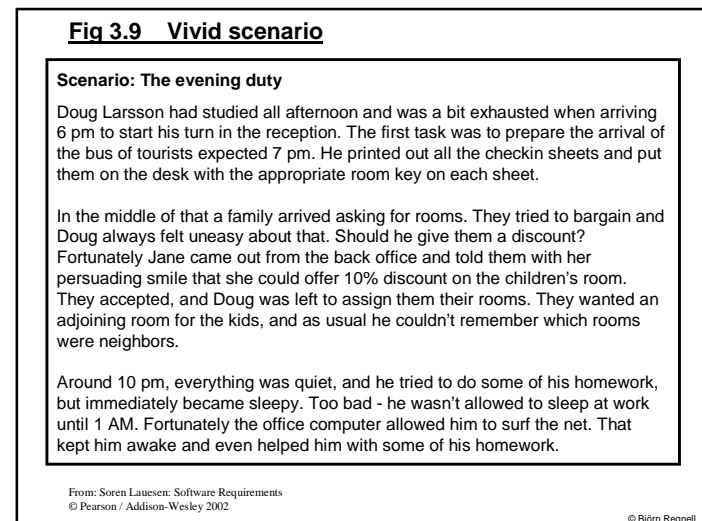


Fig 3.10 Good tasks

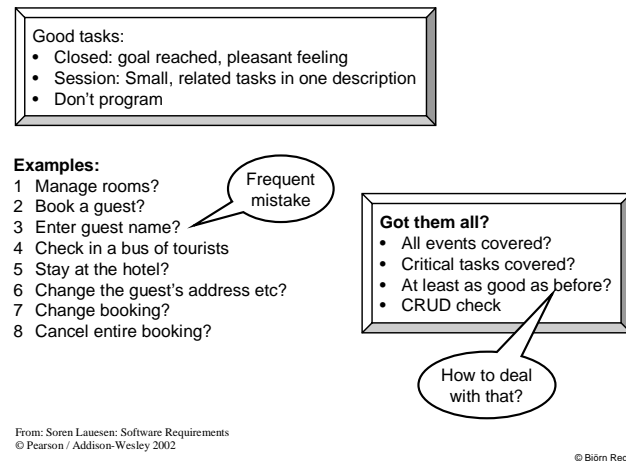


Fig 3.11 High-level tasks

Task: 1. A stay at the hotel Actor: The guest Purpose: ...	
Sub-tasks:	Example solution:
1. Select a hotel. Problem: We aren't visible enough.	?
2. Booking. Problem: Language and time zones. Guest wants two neighbor rooms	Web-booking. Choose rooms on web at a fee.
3. Check in. Problem: Guests want two keys	Electronic keys.
4. Receive service	
5. Check out Problem: Long queue in the morning	Use electronic key for self-checkout.
6. Reimburse expenses Problem: Private services on the bill	Split into two invoices, e.g. through room TV.

From: Søren Lauesen: Software Requirements
© Pearson / Addison-Wesley 2002

© Björn Regnell

Fig 3.12A Use cases vs. tasks

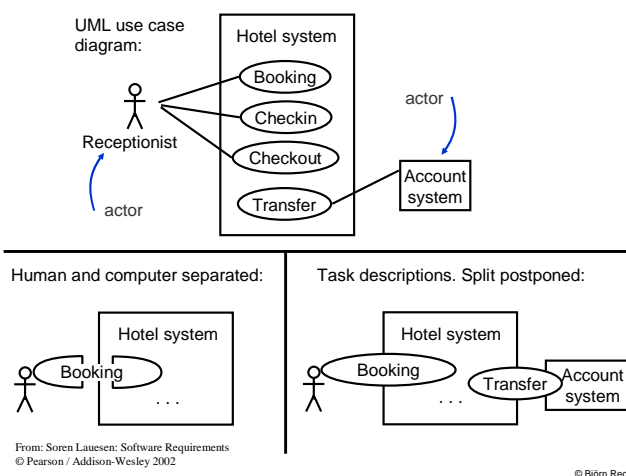


Fig 3.12B Human and/or computer

Human and computer separated	
Use case: Check in a booked guest	
User action	System action
Enter booking number	Show guest and booking details
Edit details (optional)	Store modifications
Push checkin	Allocate free room(s) Display room number(s)
Give guest key(s)	

From: Søren Lauesen: Software Requirements
© Pearson / Addison-Wesley 2002

© Björn Regnell

Computer-centric use case

Use case: Check in a booked guest

Trigger: Receptionist selects check in

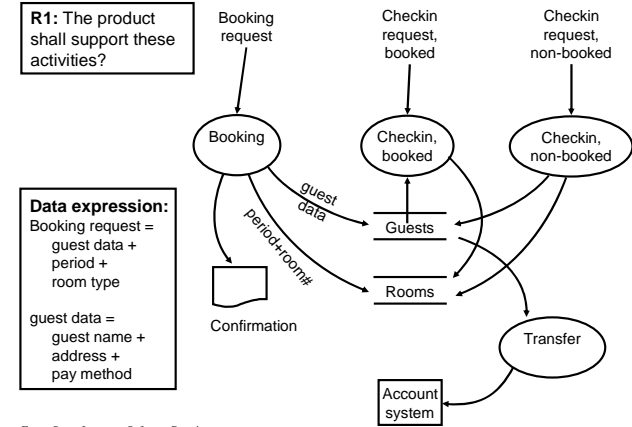
- Read booking number
- Display guest and booking details
- Read and store modifications
- Wait for checkin command
- Select free room(s)
- Mark them as occupied
- Add them to guest details
- Display room number(s)

End use case

From: Søren Lauesen: Software Requirements
© Pearson / Addison-Wesley 2002

© Björn Regnell

Fig 3.14A Dataflow - domain model



From: Søren Lauesen: Software Requirements
© Pearson / Addison-Wesley 2002

© Björn Regnell

Fig 3.15 Standards as requirements

- R1: Data transfer to the account package shall be done through a file with the format described in WonderAccount Interface Guide xx.yy. The account numbers shall be . . .
- R2: The user interface shall follow MS Windows Style Guide, xx.yy. The MS Word user interface should be used as a model where appropriate.
- R3: Shall run under MS-Windows release xx.yy. Supplier shall port product to new releases within _____ months.
- R4: Shall follow good accounting practice. The supplier shall obtain the necessary certification.
- R5: The supplier shall update the payroll computations in accordance with new union agreements within one month after release of the agreement.

From: Søren Lauesen: Software Requirements
© Pearson / Addison-Wesley 2002

© Björn Regnell

Fig 3.16 Development process as requirement

- R1: System development shall use iterative development based on prototypes as described in App. xx.
- R2: Supplier shall deliver additional screens with a complexity like screen S3 at a price of \$_____ per screen.
- R3: All developers shall spend at least two days working with the users on their daily tasks.
- R4: A special review shall be conducted at the end of each development activity to verify that all requirements and system goals are duly considered. The customer's representative shall participate in the review.
- R5: Customer and supplier shall meet at least two hours bi-weekly to review requests for change and decide what to do, based on cost/benefit estimates of the changes.

Generates new requirements?

From: Søren Lauesen: Software Requirements
© Pearson / Addison-Wesley 2002

© Björn Regnell

Functional Requirements – Summary

- **Context Diagram**
 - ♦ Diagram of product and its surrounding
 - ♦ Defining product scope
 - ♦ Very useful!
- **Event- and function lists**
 - ♦ Lists of events and functions
 - Domain or product level
 - ♦ Good as checklists at verification
 - ♦ Validation at product level?
- **Feature requirements**
 - ♦ Textual requirement: "the product shall ..."
 - ♦ High expressive power
 - ♦ Acceptable to most stakeholders
 - ♦ Can lead to false sense of security
 - How to ensure that goal-level covered?
- **Task descriptions**
 - ♦ Structured text describing user tasks
 - ♦ Easy to understand and verify
 - ♦ Good at domain level
- **(Vivid) Scenarios**
 - ♦ Rich descriptions of specific cases
 - ♦ Improves developer intuition and imagination
 - ♦ Products of elicitation but not "real" requirements
- **High-level tasks**
 - ♦ Client view of goal-related tasks
 - ♦ Independent of existing domain-level tasks
 - ♦ Good for business process re-engineering
- **Use Cases**
 - ♦ Widely used in many styles and variants
 - ♦ Some styles are good for design level (UI)
 - ♦ Can be used at different levels
 - ♦ Related to task descriptions
- **Standards as requirements**
 - ♦ Textual requirement:
 - "the product shall follow standard xxx"
 - ♦ Transfer the problem to the supplier
 - ♦ Sometimes lead to false sense of security
- **Development process requirements**
 - ♦ A requirement to follow a certain procedure
 - Use prototypes
 - Use specific reviews at certain points
 - Test in a specific way
 - Max number of simultaneous change reports
 - ...etc
 - ♦ Validation? Difficult to say how process quality relates to product quality

© Björn Regnell

Funktionella detaljer Lau:4

- Läs översiktligt så ni vet vad som finns om ni behöver det till projektet.
- Ni som läst UML kan redan delar av detta – det gäller dock att tänka på vilken nivå man använder diagrammen...

© Björn Regnell

Functional details & Special interfaces

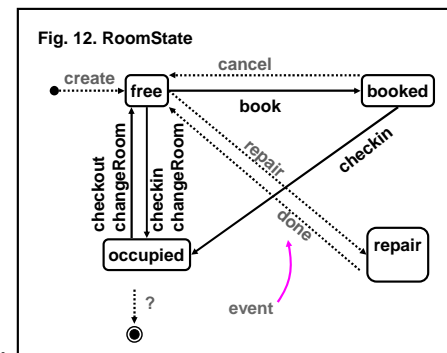
- Complex & simple functions
- Tables & decision tables
- Textual process descr.
- State diagrams**
- State-transition matrices
- Activity diagrams
- Class diagrams**
- Collaboration diagrams
- Sequence diagrams**
- Reports
- Platform requirements**
- Product integration
- Technical interfaces**

© Björn Regnell

Fig 4.4 State diagrams

Rooms have a RoomState for each day in the planning period. The status shows whether the room is free, occupied, etc. that day.

R12: RoomState shall change as shown in Fig. 12.



From: Søren Lauesen:
Software Requirements
© Pearson / Addison-Wesley 2002

© Björn Regnell

Fig 4.7A UML Class Diagram

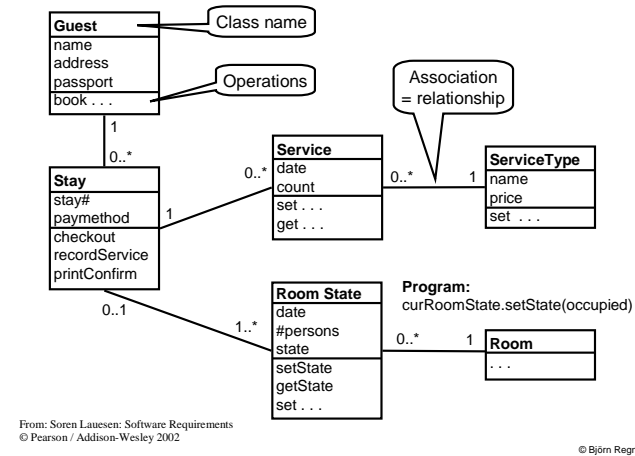
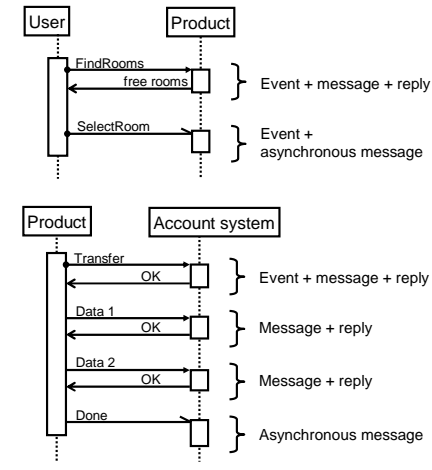


Fig 4.9 Sequence diagram



Functional details Summary

State diagrams

- Diagram showing how something changes from one state to another
- Good for finding missing functions
- Both on domain and product level
- Can sometimes be very complex and difficult to read

Class diagrams

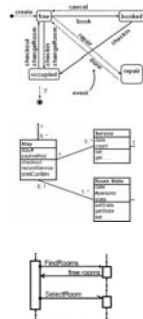
- A data model with operations on data
- Harder to understand than E/R-diagrams
- Widely used even when not good
- Not good for higher levels

Sequence diagram

- Time diagram for how objects communicate
- Good for describing (simple) communication protocols
- Useful at design-level

*Activity Diagram...

*Collaboration diagrams ...



© Björn Regnell

Vad är ett bra tentaproblem?

- Examinerar helt eller delvis ett eller flera av kursens inlärningsmål
- Kräver förståelse, inte bara gott minne
- Kopplar ihop olika delar av teorin
- Plagiat ej tillåtet; hitta på nya varianter, jmf extentor och uppgifter från tidigare år
- Det ska tydligt framgå hur tentaproblemet ska poängsättas, gärna med motivering och beskrivning var i litteraturen man kan läsa om svaret.
- Man ska inte genom att kryssa på slump ha stor sannolikhet att få poäng (typ max 20% chans annars får man även införa minuspoäng, se extentor).
- Varje inlämning ska omfatta 15p på tentan (varje rätt "krysshandling" ger ½ poäng)
- Era två inlämningar ska tillsammans ge en bra täckning av viktiga delar ur litteraturen.

© Björn Regnell

Del 1 på tentan

DEL 1. TEORI 50p

Denna del innehåller frågor som efterfrågar kryss eller bokstäver.

☐ Stämmer bra

☐ Stämmer dåligt

Cirklar i svarsalternativen avser frågor som kräver ställningstagande mellan två alternativ. Ställningstagandet anges med *ett* kryss i *en* av ringarna. Ett korrekt sätt kryss ger ½ poäng, ett felaktigt sätt kryss ger minus ½ poäng. Om inget av alternativen kryssas ges 0 poäng. Om en fråga innehåller flera ringpar poängsätts dessa var för sig. Del 1 kan totalt sett inte ge mindre än 0 poäng.

Alternativ

☐

I frågor med kvadrater i svarsalternativet efterfrågas en bokstav. Frågan anger vilka bokstäver som kan användas. T.ex. A-E för olika specificerade alternativ. Alla kvadrater ska fyllas i med exakt en bokstav. Vissa bokstäver kan förekomma flera gånger och det är inte säkert att alla bokstäver behövs. Ibland kan mer än ett alternativ vara rätt. Rätt ifyllt ruta ger ½ poäng medan felaktigt ifyllt eller ofyllt ruta ger 0 poäng.

Ni får gärna överraska med andra typer av lättträttade men förståelseinriktade frågor!

© Björn Regnell

Exempel på innovativ frågetyp: Påstående-anledning

Påstående/anledning-frågor

För varje par av påstående/anledning svara med ett av följande alternativ:

A: Både påståendet och anledningen är korrekta uttalanden OCH anledningen förklarar påståendet på ett korrekt sätt.

B: Både påståendet och anledningen är korrekta uttalanden, men anledningen förklarar inte påståendet.

C: Påståendet är korrekt, men anledningen är ett felaktigt uttalande.

D: Påståendet är felaktigt, men anledningen är ett korrekt uttalande.

E: Både påståendet och anledningen är felaktiga uttalanden.

(½ p per rätt svar)

Påstående	Anledning	Svar
Virtuella fönster passar bra till att beskriva icke-funktionella krav.	Det är lätt för kunder att med virtuella fönster validera om beskrivningar av data är fullständiga.	D
Kunderna kan med ett kontextdiagram lätt upptäcka saknade gränssnitt och diskutera vad som ska levereras.	Ett kontextdiagram ger en lättbegriplig översikt av gränssnitt och beskriver systemets avgränsning och dess aktörer.	A

© Björn Regnell

A4. Ange för varje projektyp en av dessa kundtyper som är troligast: A=en annan avdelning inom företaget som använder produkten, B=ett annat företag, C=marknadsavdelningen som ska sälja produkten.

☐ A

Egenutveckling (*in-house*)

☐ B

Anbudsfrågan (*tender*)

☐ B

Underleverantörskontrakt (*sub-contracting*)

☐ B

Inköp av hyllprogramvara (*COTS purchase*)

☐ C

Produktutveckling (*product development*)

☐ B

Kontraktsbaserad utveckling (*contract development*)

© Björn Regnell

A5. Vilka två problem ger störst skadeverkningar i praktiken? Sätt endast 2 kryss.

☐

Ofullständiga datakrav.

☒

Produkten stödjer ej användarnas arbetsuppgifter.

☒

Ofullständiga kvalitetskrav.

☐

Funktionsbeskrivningar innehåller undermåligt språk.

☐

Designkrav på standardgränssnitt.

© Björn Regnell

Inlämning av tentaproblem (ej obligatoriskt)

- Inlämning till bjorn.regnell@cs.lth.se
- Ons den 6/2 resp. Ons den 20/2 kl 23.59 om ni ska ha chans till bonus (5+5p).
- Filen ska vara i .pdf
- Döp filen enligt
<gruppbokstav>-<tenta1|2>.pdf
Attachment: A-tenta1.pdf, A-tenta2.pdf
SUBJECT: Grupp A, Tentaproblem 1

© Björn Regnell

Att göra

- Läs Lau:3.6-3.16, 4; Wei; Ka1:3.2
- Föreläsning 5 på Mån i E:C
- Övn 3 onsdag 13.15 torsdag 08.15
- Teckningslista lab 1 tillfälle 1 fre 13-17, tillfälle 2 tis 17-21
Gör labbförberedelser (finns på hemsidan)
- Senast onsdag 6/2 (Vecka4) kl 23.59:
Tentaproblemförslag 1 i .pdf till
bjorn.regnell@cs.lth.se
- Senast onsdag 20/2 (Vecka6) kl 23.59:
Tentaproblemförslag 2 i .pdf till
bjorn.regnell@cs.lth.se

© Björn Regnell