ETSN15 (2024) Requirements Engineering

Lecture 8:

- **Validation [Lau:9] & Inspections [INSP]**

  Input to project task, develop Validation Checklist for the (external) validation effort. Work on this at exercise E5.
- **Exam: Tips'n'tricks**
- **Common project challenges**
- **Info on mandatory project conference**

Björn Regnell
Elizabeth Bjarnason
http://www.cs.lth.se/krav

Requirements Validation through tests

Different types of dynamic validation:

- **Manual ”simulation” (walk-through) based on scenarios/use cases/task descriptions**
- **Paper prototypes or “mock-ups”**
- **Executable prototypes**
- **Pilot tests**

Important steps:

- **Choose suitable test approach, environment, etc.**
- **Choose who will do the testing**
- **Create & Run test cases**
- **Document problems**
- **Fix problems**
- **<u>Consider: How to avoid problems in the future?</u>**

Inspections [INSP]

Described already by
M.E. Fagan, IBM, early 70-ies

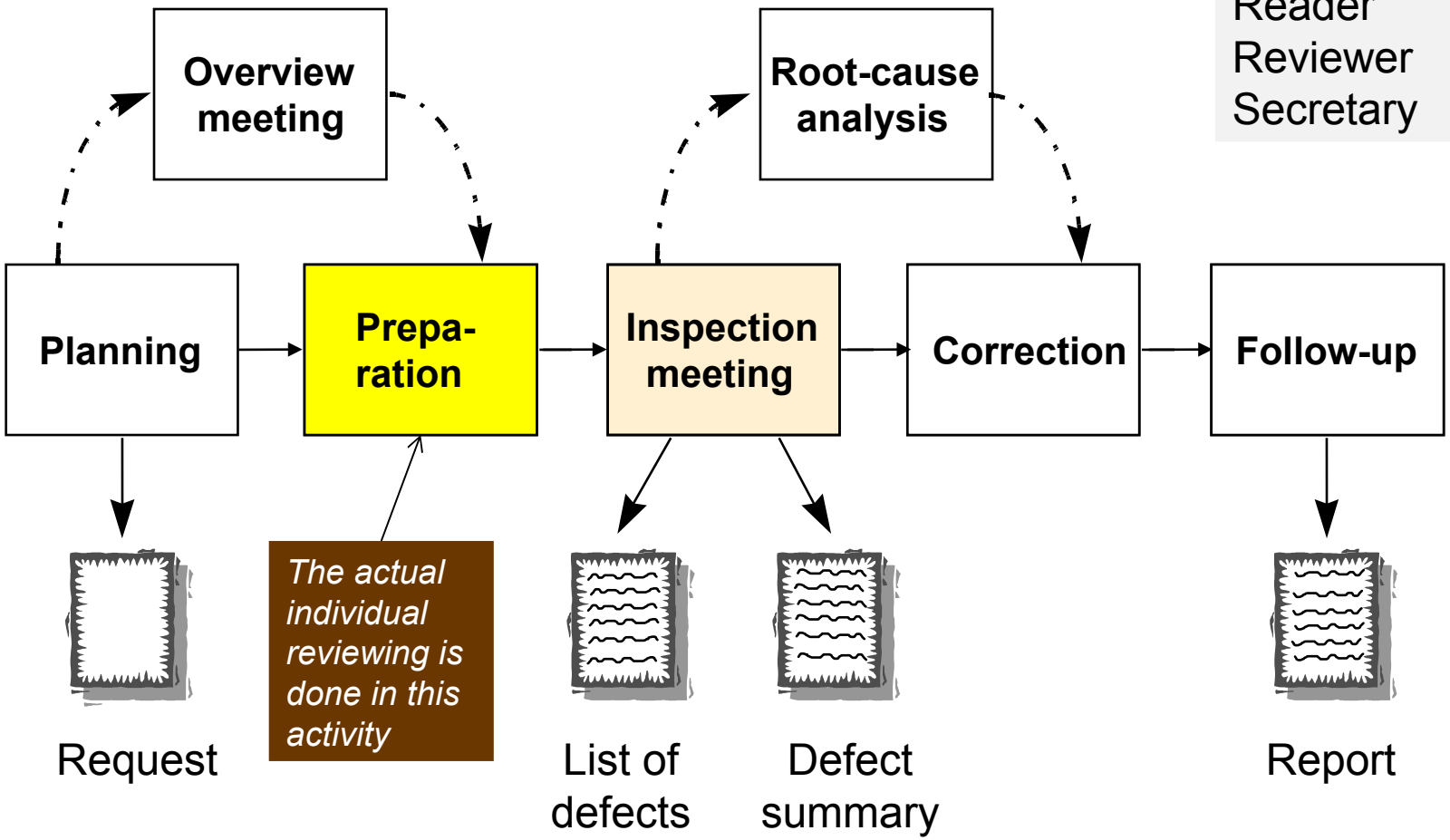
- systematic assessment
- documents inspected by others to **detect defects**

General objectives of inspection methods:

- Defect detection
- Knowledge dissemination
- Team building
- Decision-making

The inspection process [INSP]

Roles:
Author
Moderator
Reader
Reviewer
Secretary



Different methods to detect defects (reading techniques)

Ad hoc

- To your best ability (no specific guidelines)

Checklist

- A list of questions or check items direct the review

Perspective-based reading

- Different reviewers inspect from different perspectives and their findings are combined:
  e.g. user, designer, tester – perspectives,
  or from the perspective of different tasks/use cases

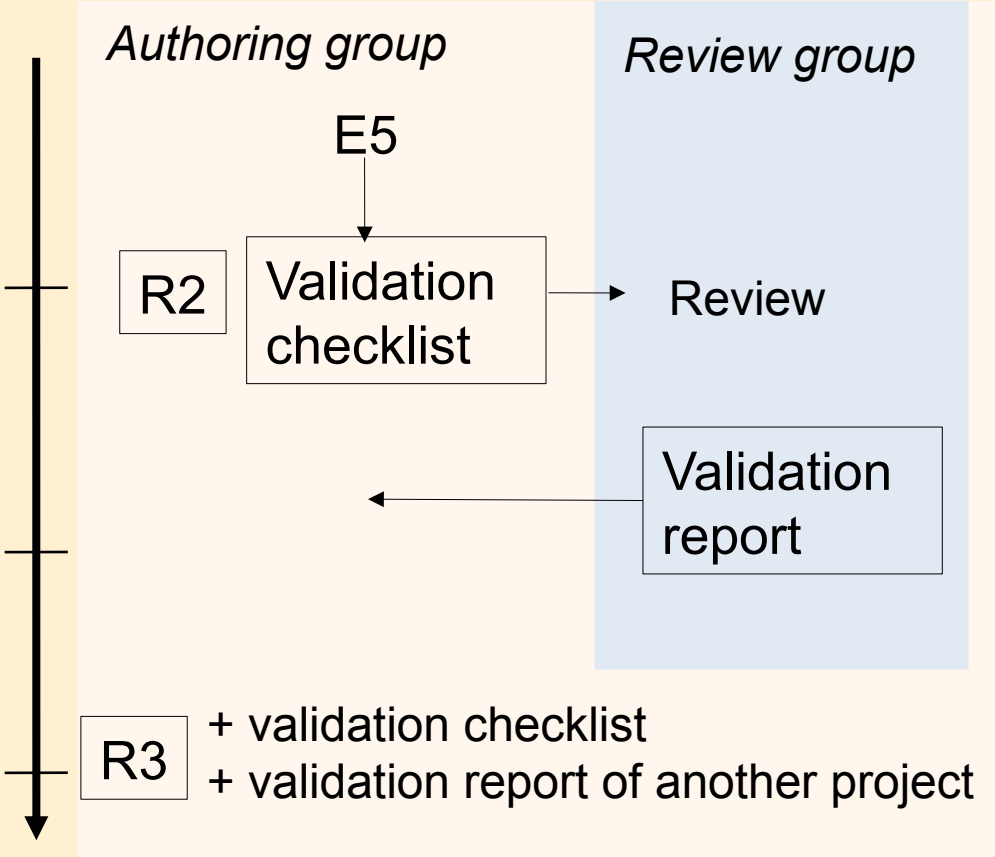
N-fold inspection

- N independent groups run inspection process in parallel

Course Project: Validation of R2 (in W6)

- Consider how to maximize value of review
- Prepare by providing the review group with a **Validation Checklist** suitable for your project **(Exercise 5!)**
- **Validation Report (**by review group) should contain relevant and useful issues ranked by criticality

See project description and Canvas assigmnents



Different kinds of checks

- Content of spec
- Structure of spec
- Consistency of spec

Fig 9.2A Contents check

Does the spec contain:

- Customer, sponsor, background
- Business goals + evidence of tracing

- Data requirements
  (database, i/o formats, comm. state, initialize)

- System boundaries & interfaces
- Domain-level reqts (events & tasks)
- Product-level reqts (events & features)
- Design-level reqts (prototype or comm. protocol)
- Specification of non-trivial functions
- Stress cases & special events & task failures

- Quality reqts (performance, usability, security . . .)

- Other deliverables (documentation, training . . .)
- Glossary (definition of domain terms . . .)

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

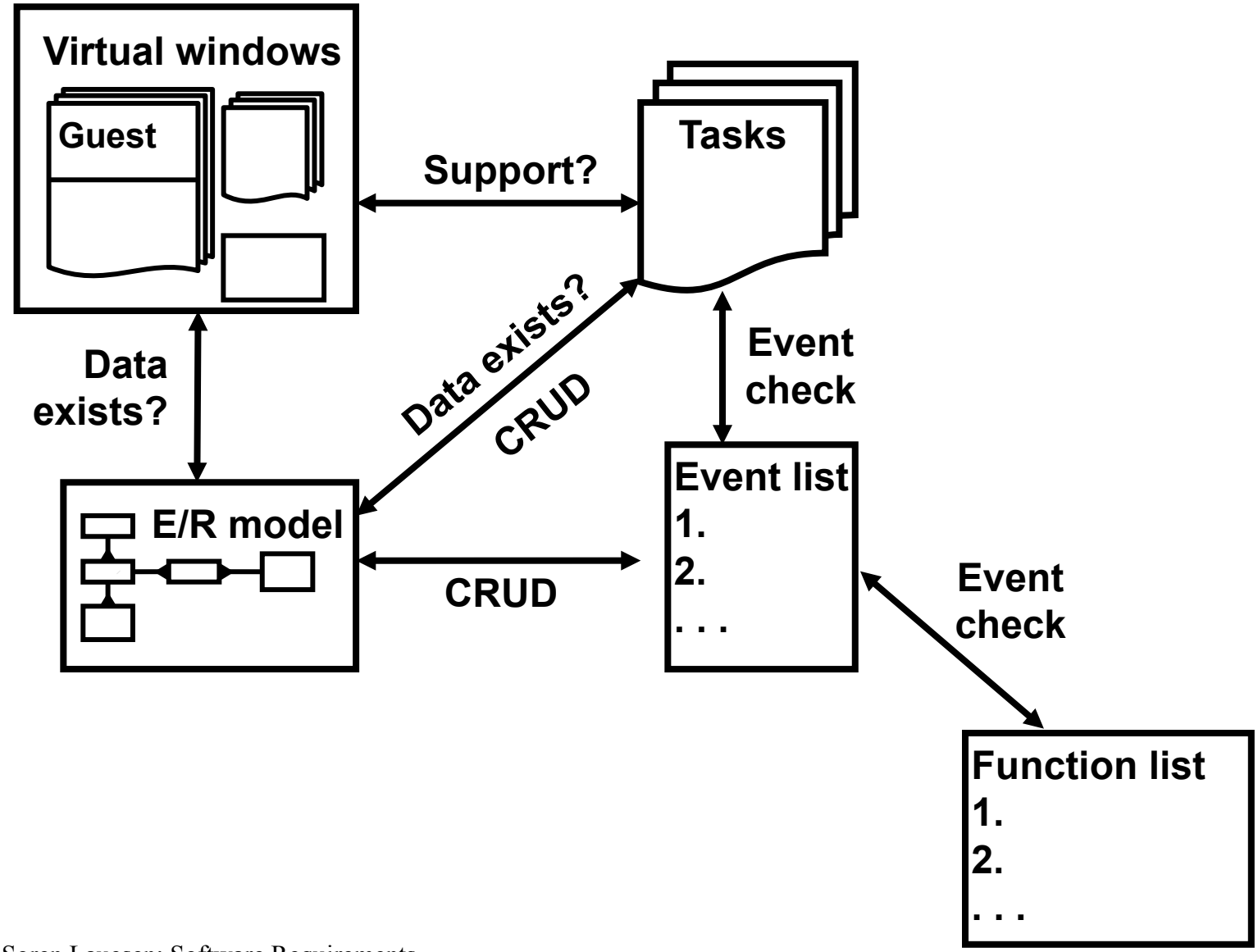
Fig 9.2B Structure check

Does the spec contain:

- ~~Number or~~ Id for each requirement
- Verifiable requirements
- Purpose of each requirement
- Examples of ways to meet requirement
- Plain-text explanation of diagrams, etc.
- Importance and stability for each requirement
- Cross refs rather than duplicate information
- Index
- An electronic version

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

Fig 9.2C Consistency checks



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

Fig 9.2D CRUD+O matrix

Create, Read, Update, Delete + Overview

Task \ Entity	Guest	Stay	Room	RoomState	Service	ServiceType
Book	C U O	C	O	U O		
CheckinBooked	RU	U O	O	U O		
CheckinNonbkd	C U O	C	O	U O		
Checkout	U	U O	R	U		
ChangeRoom	R	R	O	U O		
RecordService			O		C	R
PriceChange			C UDO			C UDO
Missing?	D	D		C?UD?	UD	

SLUT+Ö
Skapa
Läsa
Uppdatera
Ta bort
Översikt

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

Fig 9.3 Checks against surroundings

Reviews

Review:
Developers and customer review all parts.

Goal-means analysis:
Goals and critical issues covered?
Requirements justified?

Risk assessment:
Customer assesses his risk.
Developers assess their risk.
High-risk areas improved.

Tests

Simulation and walk-through
Follow task descriptions. Correct? Supported?

Prototype test (experiment with prototypes):
Requirements meaningful and realistic?
Prototype used as requirement?

Pilot test (install and operate parts of system):
Cost/benefit?
Requirements meaningful and realistic?

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

Fig 9.4(A) Check list

Project:	Noise Source Location, NSL vers. X	**Date, who:** 99-03-15, JPV
Contents check	**Observations - found & missing**	**Problem?**
Customer & sponsor	Missing, OK	
. . .		
Data: Database contents	Class model as intermediate work product	
. . .		
Initial data & states	Missing	Seems innocent, but caused many problems particularly when screen windows were opened.
Functional reqs: Limits & interfaces		
Product-level events and functions	Mostly as features	
. . .		
Special cases: Stress cases		
Power failure, HW failure, config.	Missing	**Problem.** Front-end caused many problems

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

Project:	Noise Source Location, NSL vers. X	**Date, who:** 99-03-15, JPV
Contents check (2)	**Observations - found & missing**	**Problem?**
Quality reqs: Performance	Missing, also in parts not shown here.	**Problem.** Response time became important.
Capacity, accuracy	Missing, also in parts not shown here.	**Problem.** Data volume, etc. became important.
Usability	Missing	Would have been useful
Interoperability	Missing	External dataformats, robot role, etc. caused problems
. . .		
Other deliverables: Documentation	Missing	Unimportant. Company standards exist.
. . .		

Structure check	**Observations - found & missing**	**Problem?**
ID for each req.	OK	
Purpose of each requirement	Good. Domain described.	

Consistency checks	**Observations - found & missing**	**Problem?**
CRUD check: Create, read, update, delete all data?	Have been made	

Tests	**Observations - found & missing**	**Problem?**
Prototype test	Not done, nor during development.	**Should have been done**. Caused many problems later.

[INSP] Check list

Checklist för krav		
Dokument	**Krav**	**Språk**
Finns sammanfattning? Finns författare? Finns datum? Finns innehållsförteckning? Finns alla klasser av krav? Finns definition av termer och begrepp? Finns index?	Beskriver kravet design eller ger förslag till lösningar? Beskriver flera krav samma eller liknande behov? Kan några krav grupperas ihop? Kan något krav delas upp i flera krav? Är det möjligt att uppfylla kravet med tillgänglig teknik? Är kravet unikt identifierat? Är kravet testbart? Är termer och begrepp definierade? Är kravet självständigt eller måste du undersöka andra krav för att förstå det? Kan olika personer tolka kravet på olika sätt? Har andra (liknande) krav utvärderats? Är någon information redundant? Saknas någon information?	Är alla syftningar entydiga (kolla alla "den", "det", "deras" och "dess")? Är alla komparative precisa och förståeliga (kolla alla "före", "innan", "snabbare", "efter")? Har alla ord samma betydelse för utvecklare och användare (kolla alla: "samtidigt", "kompletthet", "minst", "normalt", "i medeltal", "ofta") Innehåller något krav ord som gör kravet svårt att verifiera (kolla alla: "snabbt", "effektivt, "lagom", "minst", "mest") Finns vaga ord (kolla alla "några", "ibland", "ofta", "vanligen") Finns ofullständiga uppräkningar (kolla alla "osv.", "etc." och "till exempel")

Figur 28. Checklista för att inspektera krav.

Discussion

- What are the most important qualities of your requirements spec?
  - For commitment and planning
  - For design and testing

Fig 9.1 Quality criteria for a specification

Classic: A good requirement spec is:

Correct
Each requirement reflects a need.

Complete
All necessary requirements included.

Unambiguous
All parties agree on meaning.

Consistent
All parts match, e.g. E/R and event list.

Ranked for importance and stability
Priority and expected changes per requirement.

Modifiable
Easy to change, maintaining consistency.

Verifiable
Possible to see whether requirement is met.

Traceable
To goals/purposes, to design/code.

Additional:

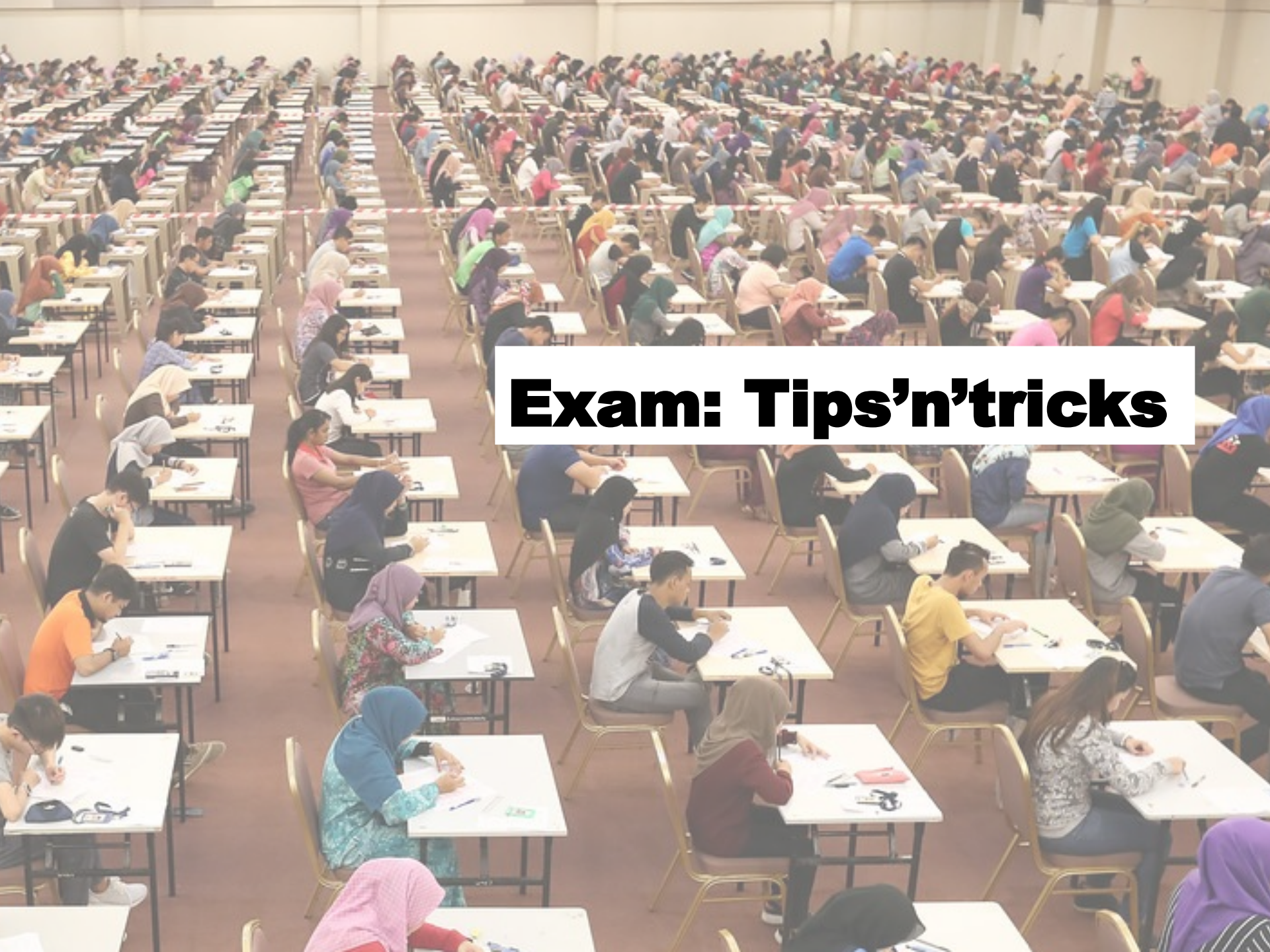
Traceable from goals to requirements.
Understandable by customer and developer.

Korrekt
Fullständig
Otvetydig
Motsägelsefri
Rankad
Modifierbar
Verifierbar
Spårbar bakåt/framåt

Begriplig

Designoberoende
Motiverad
Koncis
Välorganiserad
...

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



Exam: Tips’n’tricks

Tentamensform

- Uppgift 1: Påstående-anledning-frågor:
  krav för gk: minst 50% korrekta svar
- Uppgift 2: Essäfrågor, max 50 p
- Uppgift 3: Praktisk uppgift, max 50 p

Tentapoängen är summan av uppg 2 & 3.

Betygsberäkning:

	Project: 3	Project: 4	Project: 5
		Exam points	
Final: 3	≥ 50	≥ 50	≥ 50
Final: 4	≥ 75	≥ 67	≥ 60
Final: 5	≥ 90	≥ 83	≥ 75

U1 på Tentan: 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.

Påstående	Anledning	Svar
De studenter som deltar aktivt i undervisningsmomenten får ofta höga betyg.	Labbar o övningarna är upplagda för att stimulera praktisk tillämpning och djup-inlärning av kursens teori.	A
De studenter som deltar aktivt i undervisningsmomenten får ofta höga betyg.	Bra kommunikation med produktägare och kund minskar risken för missförstånd och felaktiga krav.	B

U1 på Tentan: 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.

Påstående	Anledning	Svar
Virtuella fönster passar bra för att beskriva icke-funktionella krav.	Virtuella fönster är en bra hjälp vid validering av fullständighet av datakrav.	
Kontextdiagram är en bra hjälp för att upptäcka saknade gränssnitt och diskutera vad som ska levereras.	Ett kontextdiagram ger en lättbegriplig översikt av systemets avgränsning och dess aktörer.	

U1 på Tentan: 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.

Påstående	Anledning	Svar
Virtuella fönster passar bra för att beskriva icke-funktionella krav.	Virtuella fönster är en bra hjälp vid validering av fullständighet av datakrav.	D
Kontextdiagram är en bra hjälp för att upptäcka saknade gränssnitt och diskutera vad som ska levereras.	Ett kontextdiagram ger en lättbegriplig översikt av systemets avgränsning och dess aktörer.	A

U2 på Tentan: Essäfrågor med nyckelord

Skriv uppsatser om ämnen + nyckelord inom max antal sidor

Uppsatserna poängsätts efter

a) hur väl **ämnet beskrivs** genom begreppen i listan, samt

b) hur väl **begreppen definieras** och **exemplifieras**.

- **Skriv läsligt**. Svårlästa eller svårbegripliga uppsatser ger avdrag.
- Börja på **nytt blad** för varje ny uppsats.

Exempel:

Ämne	**Nyckelord**
2a) Prioritering	Skalor, tekniker, intressenter, omprioritering, kravprocessen
Max 1 sida, Max 10 poäng	
2b) Elicitering	Barriärer, olika ”saker” att elicitera, eliciteringsteknikers lämplighet, intressentanalys, intervjuer, fokusgrupper, prototyper, mål-domän-analys
Max 2 sidor, Max 24 poäng	

Tips om Essäfrågor

- Gör **mindmaps** på ett kladdpapper
  - För varje nyckelord: Vad betyder det? Vilka aspekter består det av?
  - För ämnet: Hur relaterar ämnet till nyckelorden?
- Tänk ut bra **belysande exempel**, antingen ett per nyckelord, eller ett exempel som kan användas för alla nyckelord
- DÄREFTER skriv texten
  - Beskriv, **motivera**, o **exemplifiera** nyckelord OCH ämnet
  - Undvik punktlistor med ensamma ord, t ex av eliciteringsbarriärerna – ha hellre punktlistor med förklaringar av begrepp så att du visar att du förstått och kan reflektera
  - Om du har svårt för ”uppsatsskrivning”, fokusera på nyckelorden!
  - Det är inte din stilistiska förmåga som bedöms! Det ska vara lättläst och begripligt, men behöver inte vara ”språkligt vackert”.

U3 på Tentan: Praktisk uppgift

Fallbeskrivning av ett system + deluppgifter. Till exempel:

- Skapa ett **kontextdiagram** … , t ex
  - Aktörer: användarroller och andra systrem
  - Systemet under utveckling som svart låda
  - Inre domän + avgränsning yttre domän
- Gör en **intressentanalys, validera** enligt kriterier för bra krav
- **Formulera krav**
  - på **olika nivåer**, **olika typer** av krav

Uppsatserna poängsätts efter hur väl de efterfrågade koncepten (kontextdiagram, inre vs yttre domän, data krav, kravtyper etc)

a) **Realiseras**, t ex rätt aktörer, inre/yttre domängräns, formulerade krav etc

b) **Motiveras**, t ex, varför ett formulerat krav är ett målkrav

Extentor

Extentor finns på kurswebben!

https://cs.lth.se/krav

LÄS kursmaterialet i god tid!!!

Erfarenheten visar att de som går upp vid ordinarie tentatillfälle i mars klarar sig bäst!

Projekfrågor

- Praktisk dokumentstruktur?
- Användbara identiteter?
- Målnivåkrav?
- Explicita krav vs hjälp för läsaren?
- Explicita krav vs diagram och modeller?
- “Skall”, “bör”, “måste”, och prioriteter?
- Betygskriterier?

Structure of SRS: Based on Reqts Type ???

hi to lo abstraction (Top-Down) matches grading criteria

- theoretical, rather than practical
- not easy to read
- not easy to change

Structure of SRS: Based on Workarea / Feature

data reqts as overview
per feature for product & domain level, FR and QR
domain-oriented is more practical
connect context diagram to goals

Release Plan – in R3, draft in R2

Include in R2 →
Feedback prior to grading (of R3)

Unique Requirements ID:
Doc structure number + name

R: 5.2.1. ShoppingBasketDesign
The screen shown in figure 3 contains all the menu items necessary to complete any task listed in 5.2.FL. The system must provide such a screen (or something with strong resemblance).

R: 5.2.2. AddToBasket
A customer or visitor shall be able to add products from the product catalogue to a shopping basket as specified in the function list 5.2.FL.

R: 5.2.3. RemoveFromBasket
The system shall support instant removal of any and all amounts of a
from the shopping basket.

R: 5.2.4. IncrementProduct
Pressing the blue plus menu item next to each product name in as
increment by one to the amount count of the corresponding product. This shall update the amount count displayed between the blue minus/plus menu items.

R: 5.2.5. DecrementProduct
Pressing the blue minus menu item next to each product name in as shown in 3 shall decrement by one to the amount count of the corresponding product. This shall update the amount count displayed between the blue minus/plus menu items.

R: 5.2.6. ChangeAmount
The system must support chaining any amount of a product in the shopping list to a specified amount. Specified amount is from user input, inputted in the text-field between the blue minus/plus menu items. Inputted data must be any one integer ≥ 0.

Easy to locate

- Number tied to doc structure: weak modifiability

Unique Requirements ID:
Reqts Type ??? + Name

FR ReceiveGDPRResponse
The application shall have the capability
GDPR requests from companies automat

FR SendGDPRRequest
The application shall be able to formulat
of the user with the correct information

QR GDPRRequestTime
At least 90% of GDPR requests shall b
questing it.

QR RequestReplies
At least _% of data requests to compani

Reqts Type in Id
- is theoretical, rather than practical
- big risk of wrong / changing / multiple categorisation

Descriptive Name as Id:
+ provides info
+ easier to remember than number
+ independent of doc structure
+ can be found by searching

Recommendation:
Use descriptive short camel-case, single Name only!

Explicit goals

Motivate WHY the software is developed without going into details!

4.1 Product Goals

This section aims to give context to the requirements by stating the goals of the product. The requirement goals are listed below.

4.1.1 Go_ToiletInTime

Domain-level requirement

The system shall aid users getting to the bathroom in time.

4.1.2 Go_SaveTime

Implicit business goal

The system shall decrease the overall time spent by caregivers regarding incontinence.

4.1.3 Go_ReduceDiaperCost

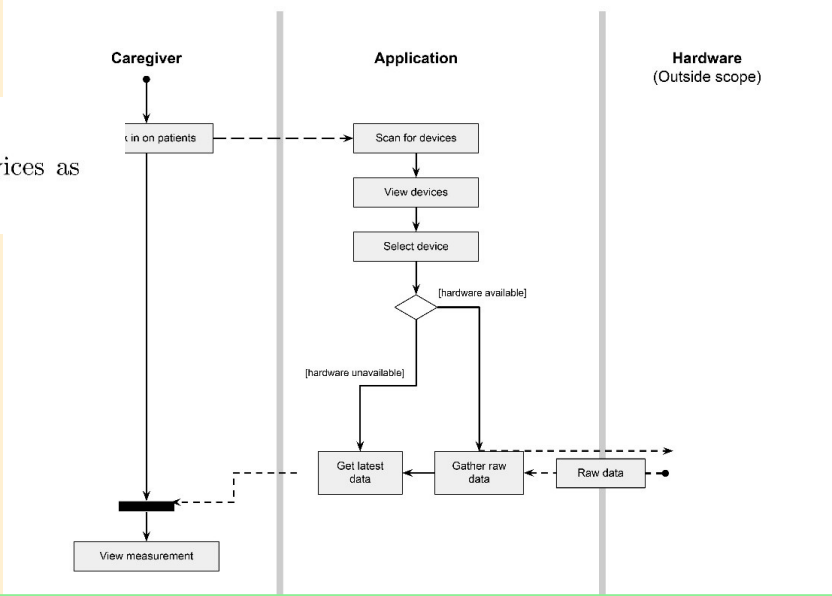
Explicit business goal

The system shall reduce the yearly cost of incontinence protection for the user with at least 50%. This is in level with what competitors is offering.

Explicit requirements vs informative text (Help for reader)

4.6.5 PrFu_CheckInActivity

The application shall support caregivers to get measurement from devices as specified in figure 11.



- All **explicit reqts** shall have a unique ID so that they are easilly spotted as a defined need/want/required condition/capability.
- Relate diagrams (context, ER, sequence charts etc) to explicit reqts by referencing them in explicit reqts, e.g. “see Figure n.n”
- Informative text helps understandability but are not explicit reqts.
- “should” (bör) indicate optionality but in an unclear way. Prefer “shall” and combine with priorities or indicate “must” when relevant.

Consistent SRS

- Between levels
  - context diagram
    - covers all reqts at domain & product level
    - for all features in context diagram - domain & product reqts
  - Domain level ↔ Product level
  - Design level → Product level
  - Product level → Design level, at least for reqts planned for R3 (in Release Plan)
- Within levels, e.g. across features at product level
- Consistent terminology

Grading: Specification

Assessment area	*Requirements for project grade* **3** Demonstrate **acceptable** ability to ...	*Also required for project grade* **4** Demonstrate **advanced** ability to ...	*Also required for project grade* **5** Demonstrate **excellent** ability to ...
Specification	**3A)** apply more than one suitable specification technique (e.g. task descriptions and screen prototypes), and more than two types of requirement (e.g. data, function, quality), and more than three abstraction levels (e.g. goal, domain, product, design). **3B)** define a system's boundaries and its interaction with external entities. **3C)** reflect on specification experiences and reason about choices of specification methods in relation to different contexts.	**4A)** combine different degrees of completeness and different levels of abstraction. **4B)** use at least four different specification techniques adequately tailored to the context. **4C)** provide explicit requirements rationale that reduce risks of misinterpretation. **4D)** use hierarchies and requirements relations to manage evolving requirements structures.	**5A)** combine specification techniques in an explicitly motivated trade-off between qualities and costs, where a high degree of specification completeness is achieved for a carefully selected subset of requirements. **5B)** provide motivated estimations of target quality levels using well-defined scales.
Elicitation	**3D)** apply more than one elicitation technique in a relevant way. **3E)** reflect on elicitation experiences.	**4E)** reason about the need for further elicitation in relation to specification quality. **4F)** demonstrate good use of prototyping to elicit realistic user requirements.	**5C)** go beyond initial stakeholders and given frames, while challenging the domain boundaries and eliciting creative ideas and deep domain knowledge in real-world contexts.
Validation	**3F)** to assess the quality of requirements and find relevant problems of several different types. **3G)** apply more than one validation technique including prototyping. **3H)** reflect on validation experiences.	**4G)** to find, prioritize and discuss requirements quality problems of different types, while reaching beyond form issues. **4H)** adapt the validation to the context and provide rationale for the chosen validation techniques.	**5D)** reason about the relation between requirements quality problems and risks, both from a product owner and developer viewpoint. **5E)** utilize links among different types of specifications in validation efforts to find and address potentially harmful inconsistencies.
Prioritization	**3I)** use more than one prioritization technique in a relevant way. **3J)** reflect on prioritization experiences.	**4I)** create a release plan for a subset of prioritized features, while taking into account precedence constraints.	**5F)** combine priorities from several stakeholders and use priorities and scheduling constraints to iteratively create a relevant release plan. **5G)** use prioritization to focus improvements of specification quality and elicitation efforts for a well-motivated subset of requirements.

Grading: Elicitation

Assessment area	*Requirements for project grade* **3** Demonstrate **acceptable** ability to ...	*Also required for project grade* **4** Demonstrate **advanced** ability to ...	*Also required for project grade* **5** Demonstrate **excellent** ability to ...
Specification	**3A)** apply more than one suitable specification technique (e.g. task descriptions and screen prototypes), and more than two types of requirement (e.g. data, function, quality), and more than three abstraction levels (e.g. goal, domain, product, design). **3B)** define a system's boundaries and its interaction with external entities. **3C)** reflect on specification experiences and reason about choices of specification methods in relation to different contexts.	**4A)** combine different degrees of completeness and different levels of abstraction. **4B)** use at least four different specification techniques adequately tailored to the context. **4C)** provide explicit requirements rationale that reduce risks of misinterpretation. **4D)** use hierarchies and requirements relations to manage evolving requirements structures.	**5A)** combine specification techniques in an explicitly motivated trade-off between qualities and costs, where a high degree of specification completeness is achieved for a carefully selected subset of requirements. **5B)** provide motivated estimations of target quality levels using well-defined scales.
Elicitation	**3D)** apply more than one elicitation technique in a relevant way. **3E)** reflect on elicitation experiences.	**4E)** reason about the need for further elicitation in relation to specification quality. **4F)** demonstrate good use of prototyping to elicit realistic user requirements.	**5C)** go beyond initial stakeholders and given frames, while challenging the domain boundaries and eliciting creative ideas and deep domain knowledge in real-world contexts.
Validation	**3F)** to assess the quality of requirements and find relevant problems of several different types. **3G)** apply more than one validation technique including prototyping. **3H)** reflect on validation experiences.	**4G)** to find, prioritize and discuss requirements quality problems of different types, while reaching beyond form issues. **4H)** adapt the validation to the context and provide rationale for the chosen validation techniques.	**5D)** reason about the relation between requirements quality problems and risks, both from a product owner and developer viewpoint. **5E)** utilize links among different types of specifications in validation efforts to find and address potentially harmful inconsistencies.
Prioritization	**3I)** use more than one prioritization technique in a relevant way. **3J)** reflect on prioritization experiences.	**4I)** create a release plan for a subset of prioritized features, while taking into account precedence constraints.	**5F)** combine priorities from several stakeholders and use priorities and scheduling constraints to iteratively create a relevant release plan. **5G)** use prioritization to focus improvements of specification quality and elicitation efforts

Grading: Validation

Assessment area	*Requirements for project grade* **3** Demonstrate **acceptable** ability to ...	*Also required for project grade* **4** Demonstrate **advanced** ability to ...	*Also required for project grade* **5** Demonstrate **excellent** ability to ...
Specification	**3A)** apply more than one suitable specification technique (e.g. task descriptions and screen prototypes), and more than two types of requirement (e.g. data, function, quality), and more than three abstraction levels (e.g. goal, domain, product, design). **3B)** define a system's boundaries and its interaction with external entities. **3C)** reflect on specification experiences and reason about choices of specification methods in relation to different contexts.	**4A)** combine different degrees of completeness and different levels of abstraction. **4B)** use at least four different specification techniques adequately tailored to the context. **4C)** provide explicit requirements rationale that reduce risks of misinterpretation. **4D)** use hierarchies and requirements relations to manage evolving requirements structures.	**5A)** combine specification techniques in an explicitly motivated trade-off between qualities and costs, where a high degree of specification completeness is achieved for a carefully selected subset of requirements. **5B)** provide motivated estimations of target quality levels using well-defined scales.
Elicitation	**3D)** apply more than one elicitation technique in a relevant way. **3E)** reflect on elicitation experiences.	**4E)** reason about the need for further elicitation in relation to specification quality. **4F)** demonstrate good use of prototyping to elicit realistic user requirements.	**5C)** go beyond initial stakeholders and given frames, while challenging the domain boundaries and eliciting creative ideas and deep domain knowledge in real-world contexts.
Validation	**3F)** to assess the quality of requirements and find relevant problems of several different types. **3G)** apply more than one validation technique including prototyping. **3H)** reflect on validation experiences.	**4G)** to find, prioritize and discuss requirements quality problems of different types, while reaching beyond form issues. **4H)** adapt the validation to the context and provide rationale for the chosen validation techniques.	**5D)** reason about the relation between requirements quality problems and risks, both from a product owner and developer viewpoint. **5E)** utilize links among different types of specifications in validation efforts to find and address potentially harmful inconsistencies.
Prioritization	**3I)** use more than one prioritization technique in a relevant way. **3J)** reflect on prioritization experiences.	**4I)** create a release plan for a subset of prioritized features, while taking into account precedence constraints.	**5F)** combine priorities from several stakeholders and use priorities and scheduling constraints to iteratively create a relevant release plan. **5G)** use prioritization to focus improvements of specification quality and elicitation efforts

Grading: Prioritization

Assessment area	*Requirements for project grade* **3** Demonstrate **acceptable** ability to ...	*Also required for project grade* **4** Demonstrate **advanced** ability to ...	*Also required for project grade* **5** Demonstrate **excellent** ability to ...
Specification	**3A)** apply more than one suitable specification technique (e.g. task descriptions and screen prototypes), and more than two types of requirement (e.g. data, function, quality), and more than three abstraction levels (e.g. goal, domain, product, design). **3B)** define a system's boundaries and its interaction with external entities. **3C)** reflect on specification experiences and reason about choices of specification methods in relation to different contexts.	**4A)** combine different degrees of completeness and different levels of abstraction. **4B)** use at least four different specification techniques adequately tailored to the context. **4C)** provide explicit requirements rationale that reduce risks of misinterpretation. **4D)** use hierarchies and requirements relations to manage evolving requirements structures.	**5A)** combine specification techniques in an explicitly motivated trade-off between qualities and costs, where a high degree of specification completeness is achieved for a carefully selected subset of requirements. **5B)** provide motivated estimations of target quality levels using well-defined scales.
Elicitation	**3D)** apply more than one elicitation technique in a relevant way. **3E)** reflect on elicitation experiences.	**4E)** reason about the need for further elicitation in relation to specification quality. **4F)** demonstrate good use of prototyping to elicit realistic user requirements.	**5C)** go beyond initial stakeholders and given frames, while challenging the domain boundaries and eliciting creative ideas and deep domain knowledge in real-world contexts.
Validation	**3F)** to assess the quality of requirements and find relevant problems of several different types. **3G)** apply more than one validation technique including prototyping. **3H)** reflect on validation experiences.	**4G)** to find, prioritize and discuss requirements quality problems of different types, while reaching beyond form issues. **4H)** adapt the validation to the context and provide rationale for the chosen validation techniques.	**5D)** reason about the relation between requirements quality problems and risks, both from a product owner and developer viewpoint. **5E)** utilize links among different types of specifications in validation efforts to find and address potentially harmful inconsistencies.
Prioritization	**3I)** use more than one prioritization technique in a relevant way. **3J)** reflect on prioritization experiences.	**4I)** create a release plan for a subset of prioritized features, while taking into account precedence constraints.	**5F)** combine priorities from several stakeholders and use priorities and scheduling constraints to iteratively create a relevant release plan. **5G)** use prioritization to focus improvements of specification quality and elicitation efforts for a well-motivated subset of requirements.

Project conference
Wed W7 come <u>15:10 latest</u>

- Submit presentation material in pdf in Canvas by **Wednesday 08.00**
- Max **9 minutes** presentation; will be interrupted!
- Presentation outline
  - ~ 1 minute about project mission
  - ~ 4 minutes overview of project results
  - ~ 4 minutes about methods and experiences
- Questions by **discussant group** (same as validation group) – max **4 min**
  - E.g. choice of RE techniques, experienced RE challenges & solutions
- Max **1 minute** for switching to next group
- One or max 2 presenters (not too much time on switching)
- All group members contribute in developing/rehearsing the presentation
- **Practice before** to keep time and focus on the most important!
- If you want to practice English this is a good chance! (Swedish is also OK)

Order of presentation at Project Conference

Mandatory attendance!

Presenter	Discussant
Codiska	Pedatim
Pedatim	M-Solar
M-Solar	Vomerce
15 min break	
Vomerce	Codiska

To do ...

- Read Lau:9, [INSP]
- Exercise E5 Validation: project validation preparation
  bring your **System Requirements specification + text book (Lauesen)**

Week 6

- Project deliverables (see Project description + Course program):
  - Release R2 & **Validation checklist** (Sun 23.59)
  - **Validation Report** based on checklist from other group (Thu)
    - Submit pdf via Canvas AND Send also via Canvas Inbox to other group (SRS R2 + Checklist + Validation Report)
- Project meeting with supervisor
- Sign-up for exam (**tentaanmälan** öppnar på måndag!)

Week 7

- Submit **Conf Presentation MATERIAL (CP) Wed W7 before 8.00** hrs
- Prepare questions as discussant group
- Wed W7 be there **15.10** latest for PROJECT CONFERENCE
  Projektkonferensen är exmamination med obligatorisk närvaro!
  Giltiga frånvaroskäl tex sjukdom meddelas bjorn.regnell@cs.lth.se
  Annan individuell examination tilldelas frånvarande vid giltiga skäl.