



LUND
UNIVERSITY

ETSN15: Exercise 5

Validation [INSP, Lau:9]

ELIZABETH BJARNASON



Why bother validating & checking?



Ensuring “fit”

- To needs – customers, users?
- Aligned with company roadmaps & directions?
- Technically feasible – realistic requirements?
- Fit with laws and regulations?

Does requirements specification enable

- Building the right system?
- Testing and maintaining the system?
- Managing later changes?

Methods – static vs dynamic

- Inspections / Doc reviews [INSP, Lau:9]
- Tests, e.g. usability testing, prototypes, model-based simulations [Lau:9]





Requirements validation

Purpose to ensure

that we have elicited and documented the right requirements in a good way

- ◆ Will we **build the right system** with these requirements?
- ◆ Do these reqts provide **sufficient information** for testing?
- ◆ **Correct info** communicated & promised to customers?
- ◆ **Support for managing** later requirements changes?

Methods

- ◆ Inspections / Doc reviews [INSP, Lau:9]
- ◆ Tests, e.g. usability testing, prototypes, model-based simulations [Lau:9]



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



Requirements Validation through tests

Different types of **dynamic validation**:

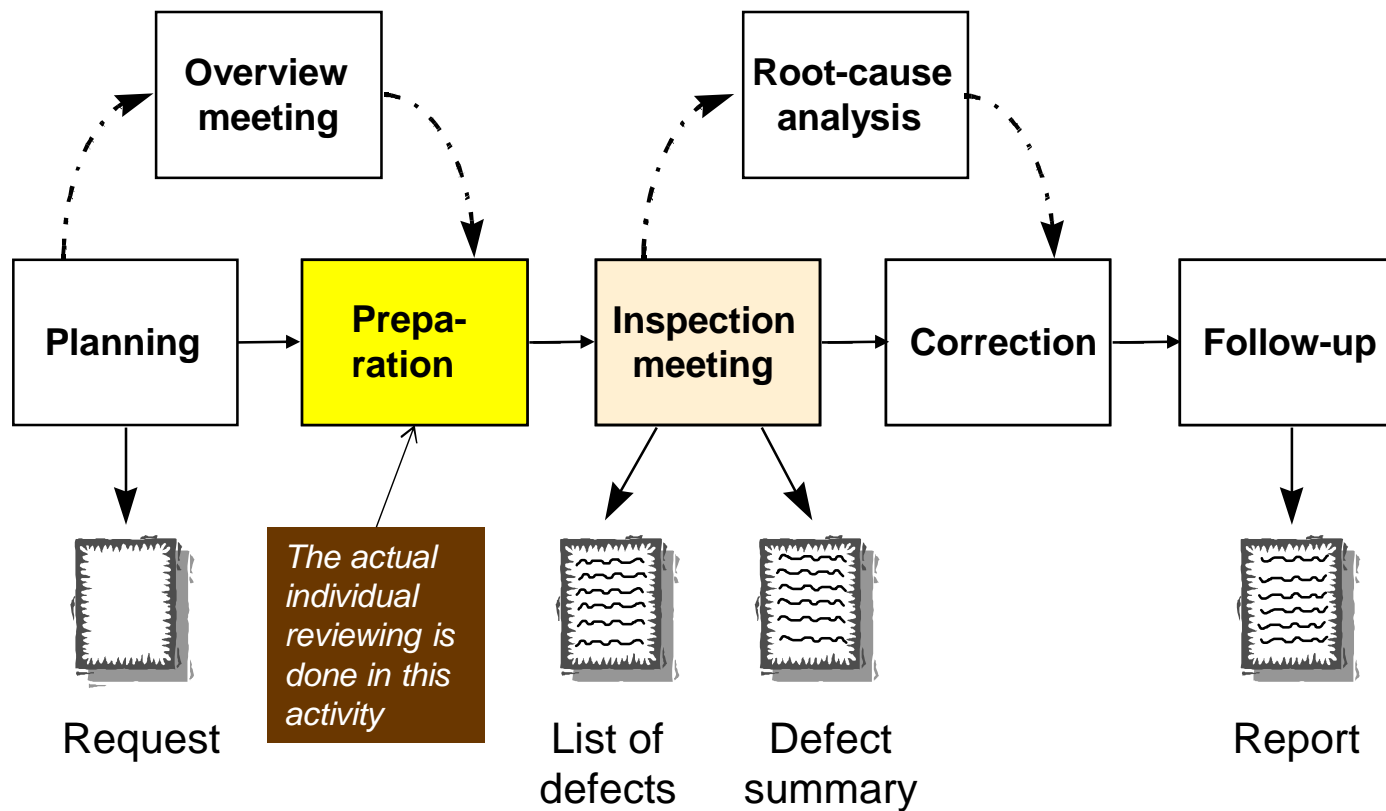
- ◆ **Manual "simulation" (walk-through) based on scenarios/use cases/task descriptions**
- ◆ **Paper prototypes, "mock-ups", executable prototypes**
- ◆ **Pilot tests**

Important steps:

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

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

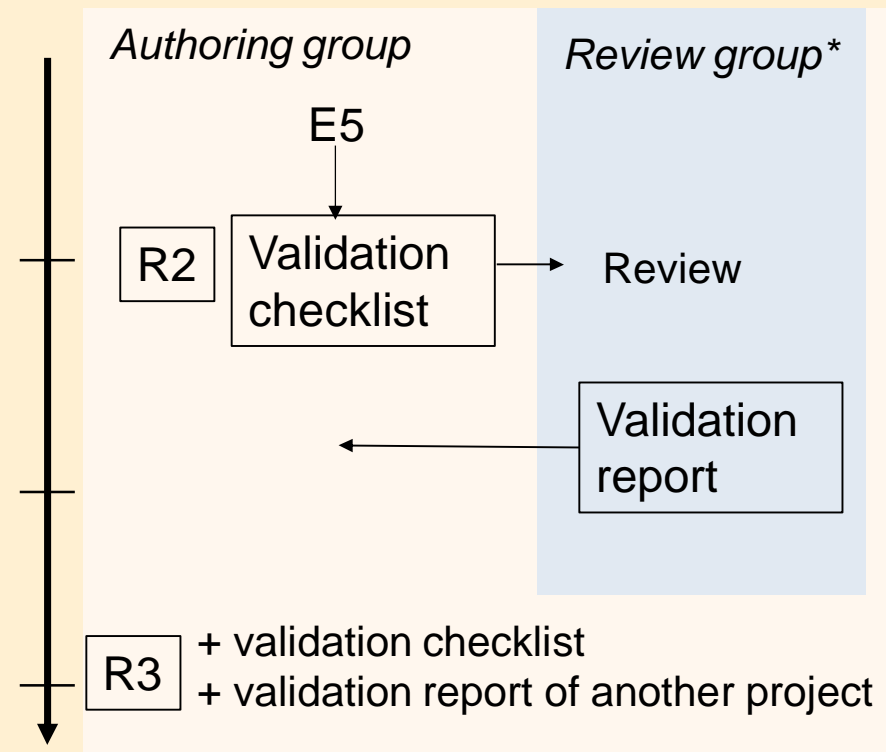
N-fold inspection

- ◆ 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

**Review groups,
see Canvas – Checklist hand-in*



Hand-in and send Validation check-list

Due Sunday by 11:59pm Points 0 Submitting a file upload

Do the following

- Submit your validation check list here (in Canvas)

AND

- Send via Canvas Inbox your validation checklist + SRS for R2 to the group that is to review your requirements.

NOTE: The reviewing group is also responsible for asking questions on the authoring group's oral presentation at the final project conference. For example, ask about choice of RE techniques, experienced RE challenges & solutions during the project, etc. **Keep this in mind while reviewing their SRS!**

| Authoring group | Reviews and Discusses (at Project Conference) THIS group |
|-----------------|--|
| 1 Codiska | 2 Pedatim |
| 2 Pedatim | 3 M-Solar |
| 3 M-Solar | 4 Vomerce |
| 4 Vomerce | 1 Codiska |

Each project group

- reviews the group AFTER them in the list provided in the table above, AND,
- sends their SRS and checklist to the group BEFORE them in the list

For example, group 1 Codiska is to

- review and write validation report for group 2 Pedatim's SRS, and
- send their SRS and checklist to group 2 (left-hand column in table above).

While group 2 is to

- review and write a validation report for group 1's SRS, and

Your two roles in validation

- As **author** make a useful **checklist**
- As **reviewer** make a useful **validation report**

Group 1 reviews Group 2's R2 SRS etc

Also look at grading criteria for Validation

Different kinds of checks [Lau:9]

- Content of spec
- Structure of spec
- Consistency of spec
- Checks against surroundings:
Review & Tests



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

Fig 9.2A Contents check

| |
|---|
| Does the spec contain: <ul style="list-style-type: none">• Customer, sponsor, background• Business goals + evidence of tracing |
| <ul style="list-style-type: none">• 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 |
| <ul style="list-style-type: none">• Quality reqts (performance, usability, security . . .) |
| <ul style="list-style-type: none">• Other deliverables (documentation, training . . .)• Glossary (definition of domain terms . . .) |



This Photo by Unknown Author is licensed under [CC BY-SA](#)

Fig 9.2B Structure check

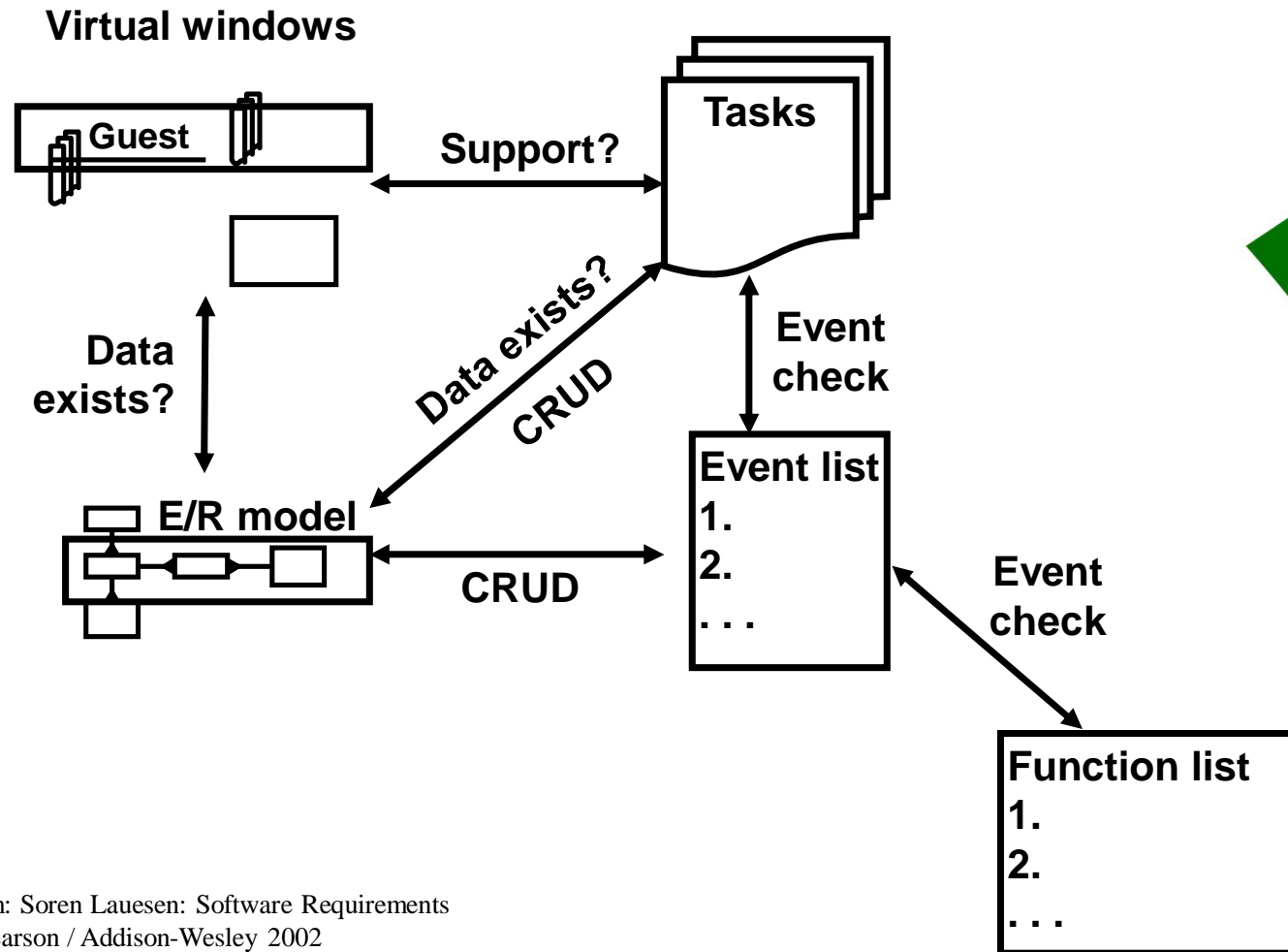
Does the spec contain:

- 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



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

Fig 9.2C Consistency checks



This Photo by Unknown Author is licensed under [CC BY-SA](#)

Fig 9.2D CRUD+O matrix

Create, Read, Update, Delete + Overview

| Entity \ Task | 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



This Photo by Unknown Author is licensed under [CC BY-SA](#)

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?



Discussion



What are the quality criteria for a requirements specification?

- For contractual purposes
- For planning purposes
- For development
- For testing



Criteria for Good Requirements

IEEE 830 Standard



Correct

Incorrect requirements are useless and potentially dangerous!
If the requirements are not correct, we risk spreading misinformation within project and to customers.

Complete

Spec covers all necessary requirements to describe the full scope incl. exceptions, error handling etc

Unambiguous

Everyone understands it the same way. Can everyone read, discuss + agree on what it means?

Clear & Concise

Simply and clearly stated. Makes it easier for others (incl pure readers) to understand.





Consistent

Are there requirements that contradict each other?

Modifiable

Modifications are easy to make, maintaining consistency of the whole specification



Verifiable

If a requirement is not verifiable, determining whether it was correctly implemented is a matter of opinion.

Design independent

Requirement describes functionality from user perspective, not how to implement

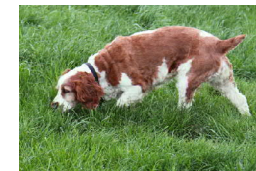


Ranked for importance and stability

Info needed to handle changes; why is req important (reqts motivation / prio / stakeholder), likely to change?

Traceable

What motivates this reqt? Indicates if it is needed. Useful when discussing scope &/ reqts changes.



Example 1

Shut off the pumps if the water level remains above 100 meters for more than 4 seconds.

Correct
Complete
Unambiguous
Clear & Concise
Consistent
Ranked
Modifiable
Verifiable
Traceable
Design independent

Example 1

Shut off the pumps if the water level remains above 100 meters for more than 4 seconds.

Shut off in which scenario?

- When MIN(water level) > 100 m?
- When MAX(water level) > 100 m?
- When AVERAGE(water level) > 100 m?

For which time period?

- Continuous period of 4 s?
- Summed up time for “too high level”?

Correct

Complete

Unambiguous

Clear & Concise

Consistent

Ranked

Modifiable

Verifiable

Traceable

Design independent

Example 2

Aircraft that are non-friendly and have an unknown mission or the potential to enter restricted airspace within 5 MINUTES shall raise an alert.

Correct
Complete
Unambiguous
Clear & Concise
Consistent
Ranked
Modifiable
Verifiable
Traceable
Design independent

Example 2

*Aircraft that are non-friendly **and** have an unknown mission **or** the potential to enter restricted airspace within 5 MINUTES shall raise an alert.*

What is meant by

- non-friendly? Unknown mission – to whom? Etc.

How tie and/or?

(NF and UM) or (RA within T)

(NF) and (UM or (RA within T))

Etc...

Who shall raise the alert –the non-friendly aircraft?

Correct

Complete

Unambiguous

Clear & Concise

Consistent

Ranked

Modifiable

Verifiable

Traceable

Design independent

Example 3

Create a means for protecting a small group of human beings from the hostile elements of their environment.

Correct
Complete
Unambiguous
Clear & Concise
Consistent
Ranked
Modifiable
Verifiable
Traceable
Design independent

Example 3

Create a means for protecting a small group of human beings from the hostile elements of their environment.

What is the context, e.g.

- Astronauts on Mars
- Tourists caught in a snowstorm
- Sailors that fall overboard

How “small” is small?

Correct

Complete

Unambiguous

Clear & Concise

Consistent

Ranked

Modifiable

Verifiable

Traceable

Design independent

Example 4

The product shall provide status messages at regular intervals not less than every 60 seconds.

Correct
Complete
Unambiguous
Clear & Concise
Consistent
Ranked
Modifiable
Verifiable
Traceable
Design independent

Example 4

The product shall provide status messages at regular intervals not less than every 60 seconds.

- What status messages?
 - How displayed to user?
 - Should interval between message be “more than every 60 s”?
- Avoid NEGATIVE requirements**

Correct
Complete
Unambiguous
Clear & Concise
Consistent
Ranked
Modifiable
Verifiable
Traceable
Design independent

Example 5

The product shall switch between displaying and hiding non-printing characters instantaneously.

Correct
Complete
Unambiguous
Clear & Concise
Consistent
Ranked
Modifiable
Verifiable
Traceable
Design independent

Example 5

The product shall switch between displaying and hiding non-printing characters instantaneously.

- Instantaneously is not feasible or correct!
- What triggers the switching? The user? Some condition?
- Scope of the change within document – selected text, all text, something else?

Correct

Complete

Unambiguous

Clear & Concise

Consistent

Ranked

Modifiable

Verifiable

Traceable

Design independent

Example 6

The HTML parser shall produce an HTML markup error report which allows quick resolution of errors when used by HTML novices.

Correct
Complete
Unambiguous
Clear & Concise
Consistent
Ranked
Modifiable
Verifiable
Traceable
Design independent

Example 6

The HTML parser shall produce an HTML markup error report which allows quick resolution of errors when used by HTML novices.

- How quick is quick? Unquantified
- What info in error report?
- How define HTML novice?

Correct
Complete
Unambiguous
Clear & Concise
Consistent
Ranked
Modifiable
Verifiable
Traceable
Design independent

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 |

| | | |
|---|---|--|
| 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. |

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

Figur 28. Checklista för att inspektera krav.

Today's exercises

- Who can perform the checks? – 1
- IEEE criteria for different types of checks – 2
- Design your own validation checklist – 3



LUND
UNIVERSITY