ETSN15: Exercise 5
Validation [INSP, Lau:9]

ELIZABETH BJARNASON

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?
- Conveying correct info to customers?
- 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 through tests

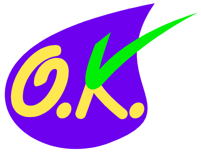
Different types of **dynamic validation**:

- **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?**

Criteria for Good Requirements IEEE 830 Standard



Correct

Incorrect requirements are useless and potentially dangerous! If the requirements are not correct, we risk spreading mis-information 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.

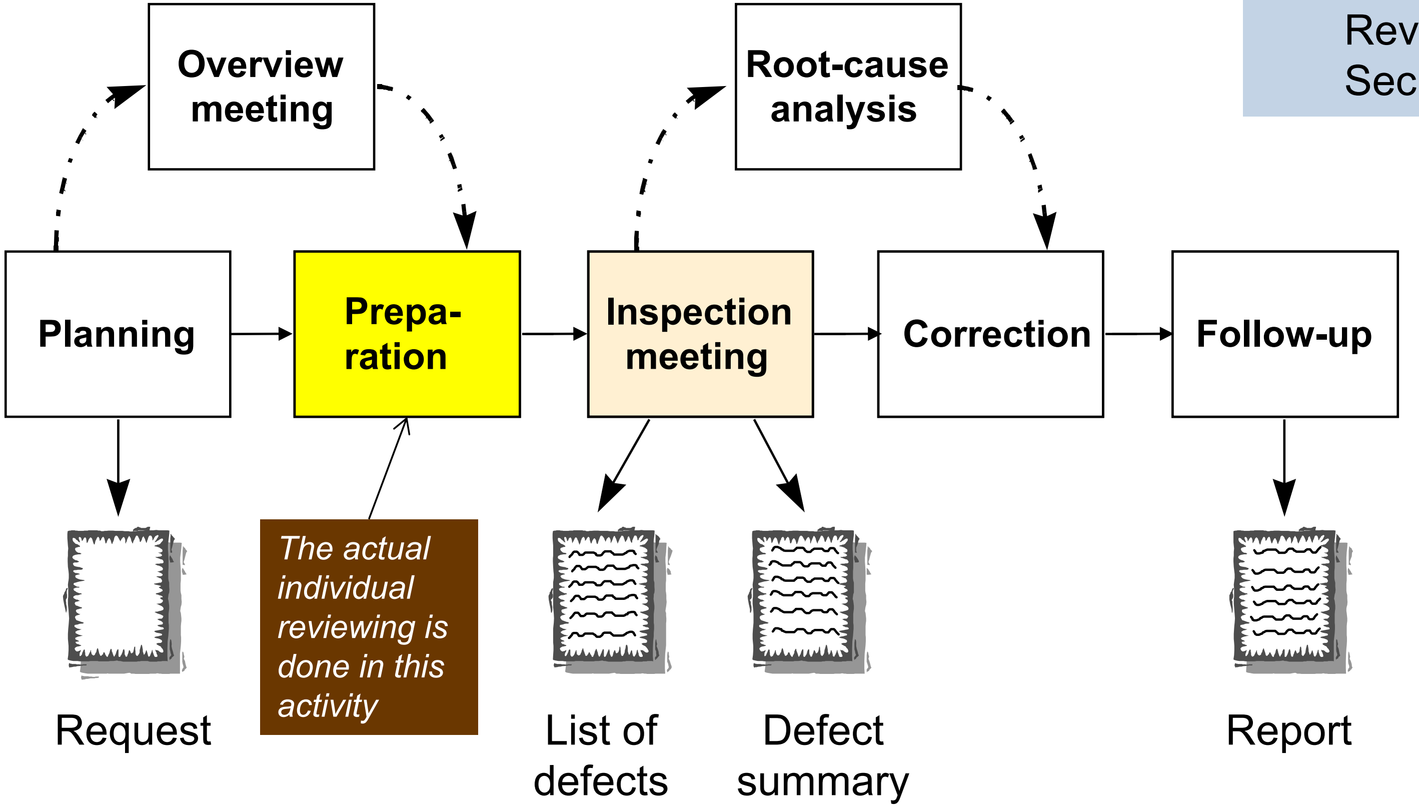
SHALL, SHOULD, and MUST

TERM	REQUIREMENT LEVEL	FLEXIBILITY	TYPICAL USAGE
MUST	Absolute requirement	No deviation allowed	Critical functions, safety, security
SHALL	Mandatory requirement	Minimal to none	Contractual, formal specifications
SHOULD	Recommended requirement	Deviation permitted w justification	Best practices, enhancements

The inspection process [INSP]

Purposes:
Defect detection
Knowledge sharing
Team building
Decision-making

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

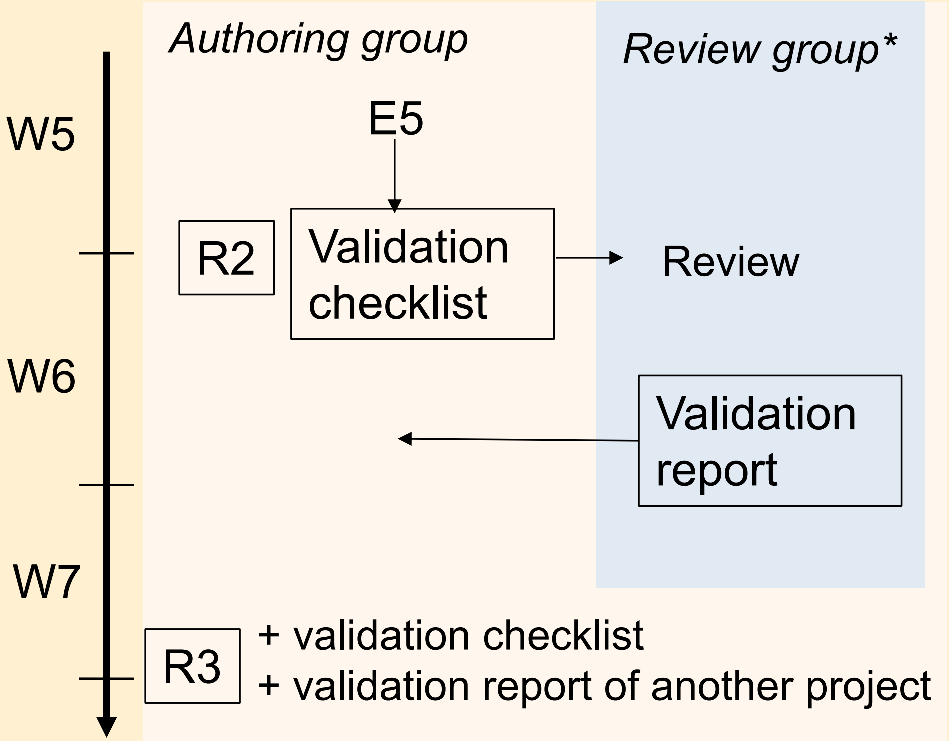
N-fold inspection

- N independent groups run inspection process in parallel

Course Project: Validation of R2 (in W6)

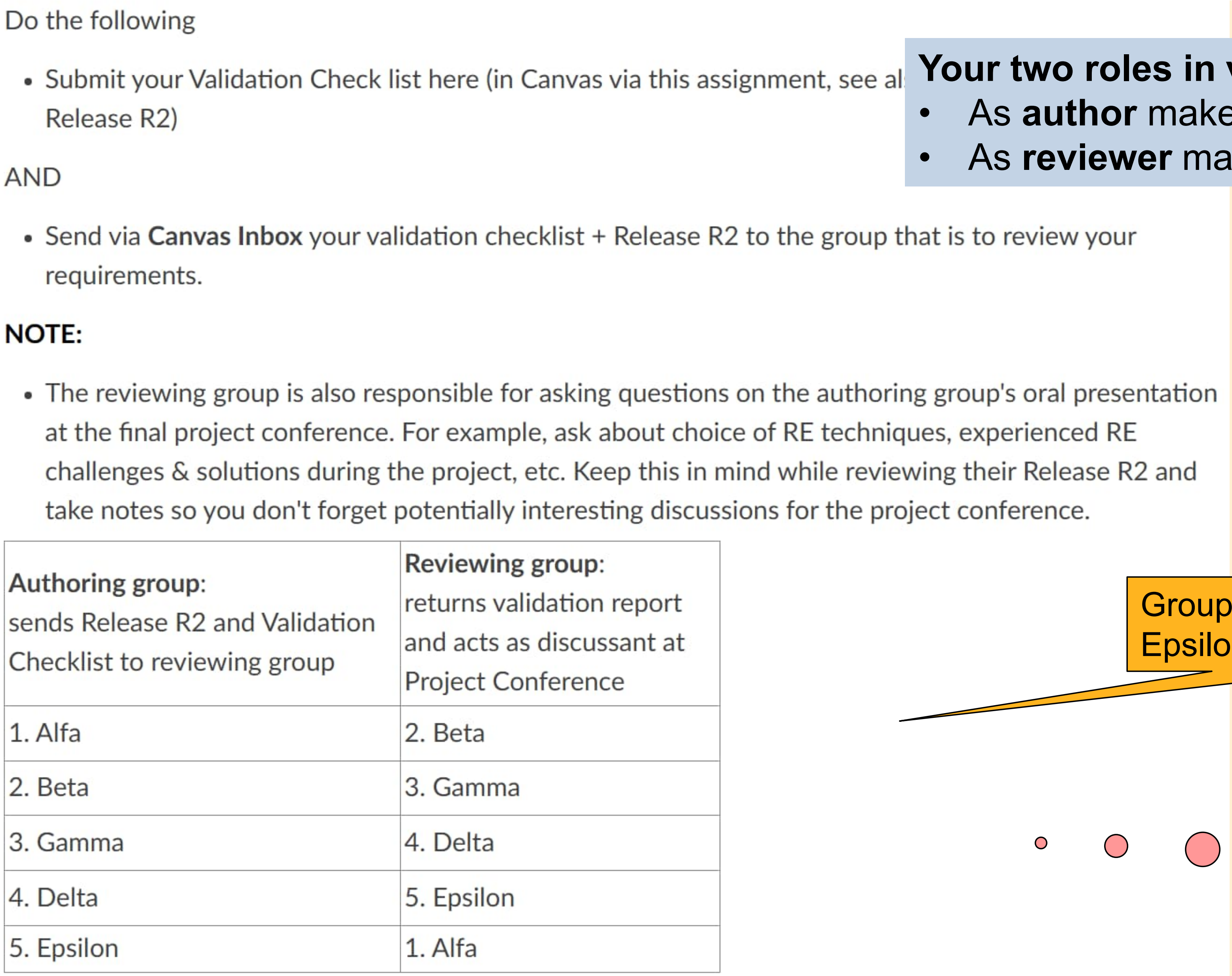
- Consider how to maximize value of review
- Prepare by providing the review group with **your R2** and a **Validation Checklist** suitable for your project **w5.7**
- **Validation Report (**by review group) should contain relevant and useful issues ranked by criticality **w6.4**

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



Your two roles in validation

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



Do the following

- Submit your Validation Check list here (in Canvas via this assignment, see al Release R2)

AND

- Send via **Canvas Inbox** your validation checklist + Release 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 Release R2 and take notes so you don't forget potentially interesting discussions for the project conference.

Authoring group: sends Release R2 and Validation Checklist to reviewing group	**Reviewing group**: returns validation report and acts as discussant at Project Conference
1. Alfa	2. Beta
2. Beta	3. Gamma
3. Gamma	4. Delta
4. Delta	5. Epsilon
5. Epsilon	1. Alfa

Group Alfa reviews Group Epsilon's Release R2

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

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

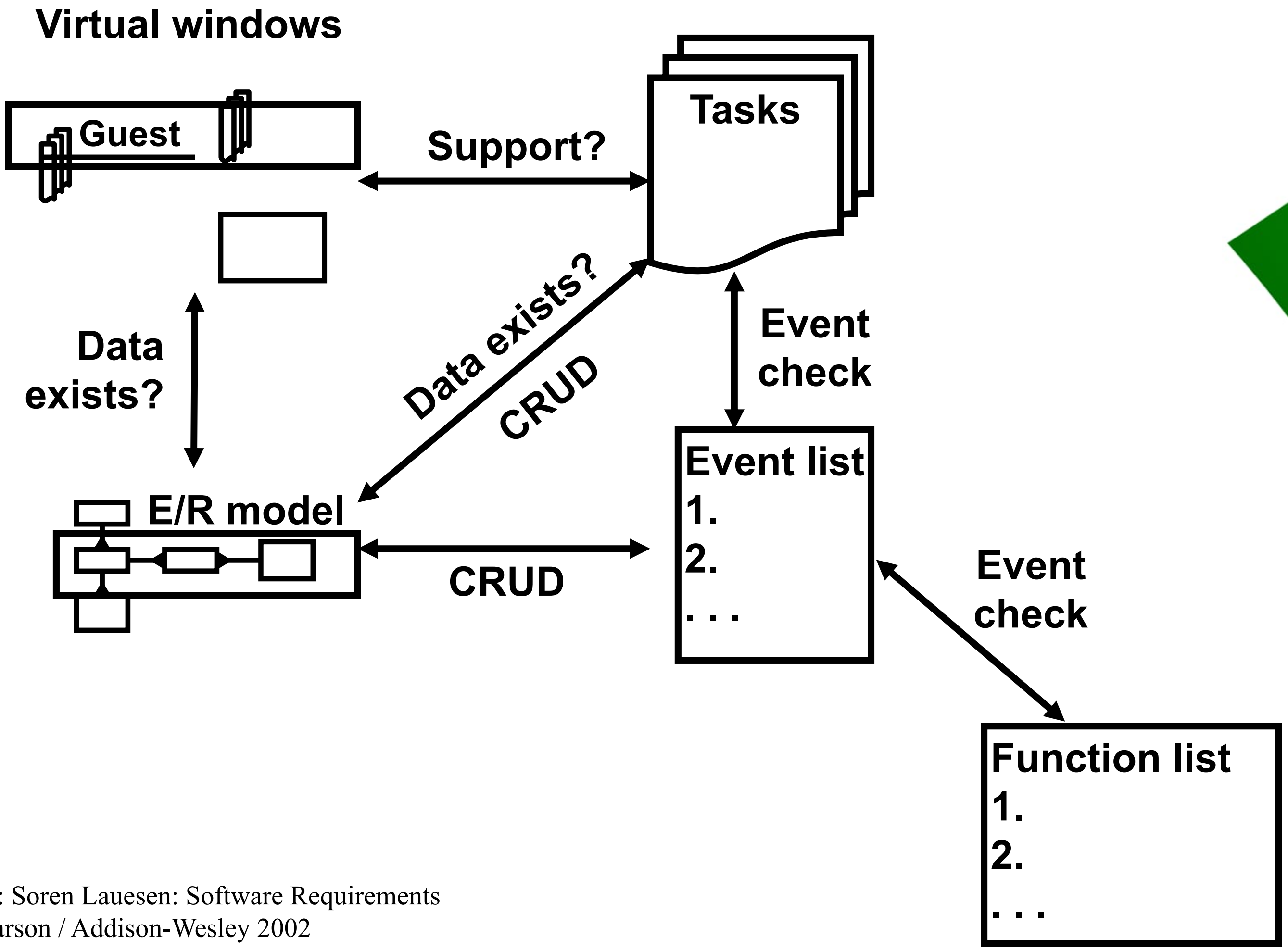
Fig 9.2B Structure check

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

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

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

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

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

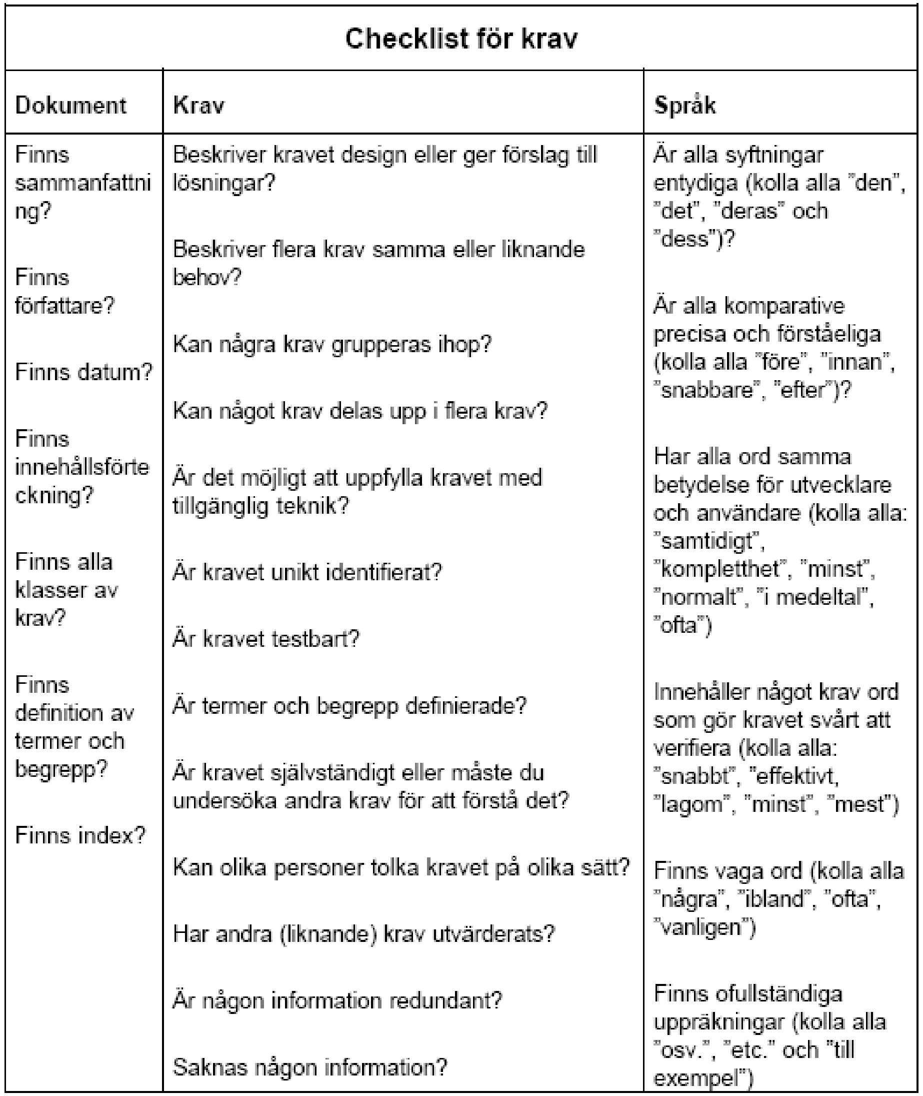
From: Soren Lauesen: Software Requirements

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? 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.