# ETSN15 Kravhantering

## Lecture 2:

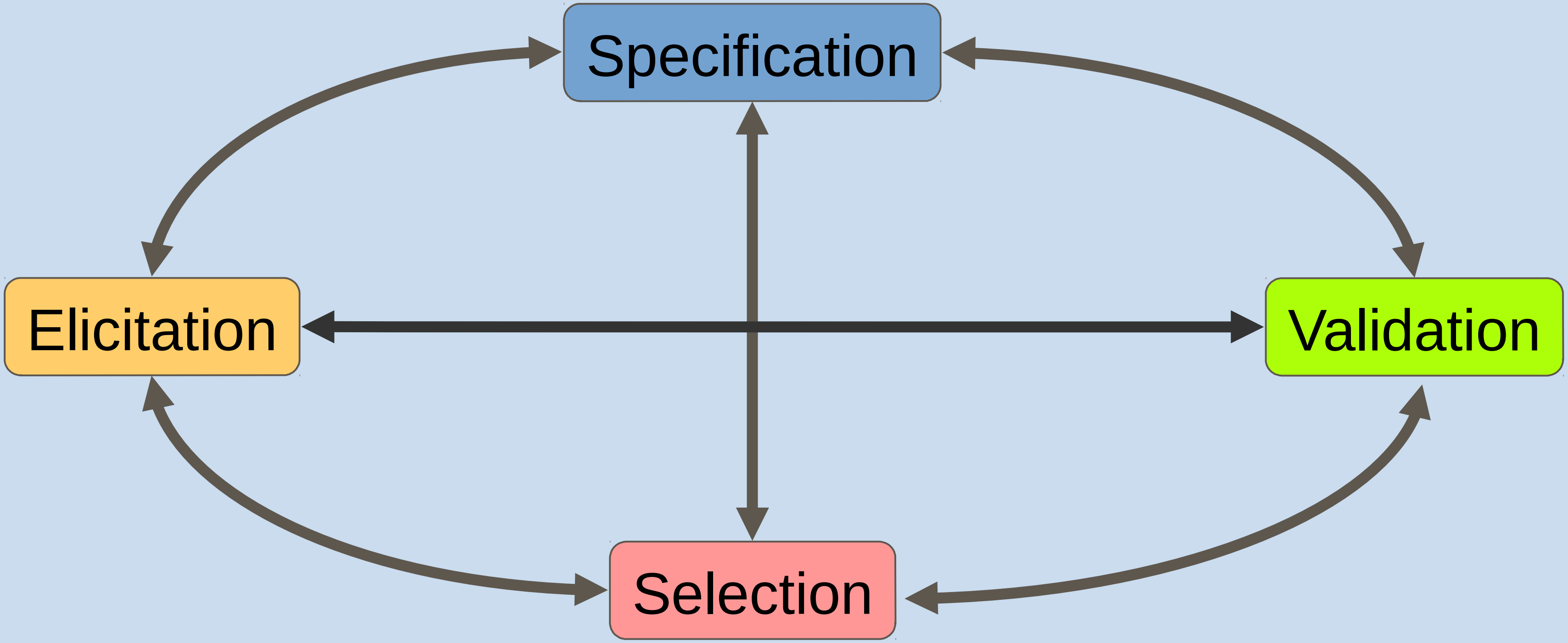
**Project Missions**

**Elicitation:  Lau:8**
**Prioritization: [PRIO]**

**Course ombudsman volunteers**

**Björn Regnell**
http://www.cs.lth.se/krav

# In practice you need to iterate and work in parallel



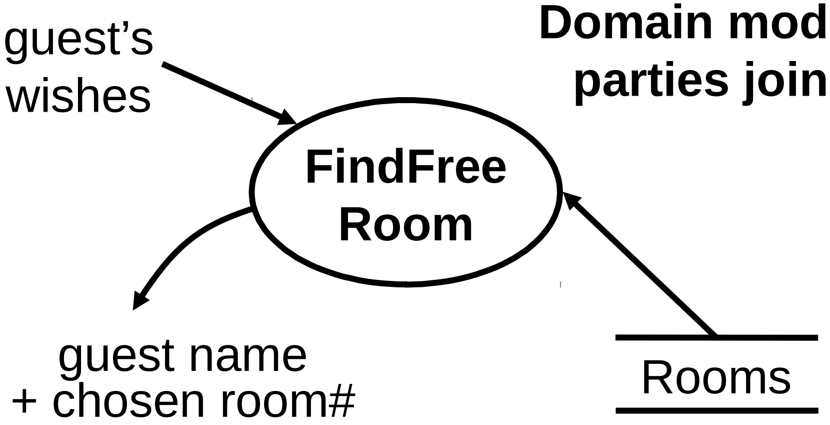
# Different project types

- **In-house** – Internutveckling för egna behov
- **Product Development** – Produktutveckling för öppen marknad, t.ex. inbyggda system, generella appar för en marknad (COTS develpment), etc.
- **Time & Materials** – Utveckling på löpande räkning, rörligt pris
- **Commercial Off-The-Shelf software purchase** (COTS purchase) – Inköp av generisk (hyll-) programvara
- **Customization** – Kundspecifik anpassning av generisk programvara
- **(Request for) Tender** – Anbudsförfrågan
  - Customer specific: för upphandling av kundspecifik utveckling
  - Generic (COTS): för upphandling av generisk programvara
- **Contract development** – Kontraktsbaserad utveckling med fast/rörligt pris
- **Sub-contracting** – Underleverantörskontrakt med fast/rörligt pris
- **Unknown**, pre-study – Okänd, förstudie för att utreda lämplig projekttyp
- **Hybrid** – kombinationer av ovanstående
- ... ?

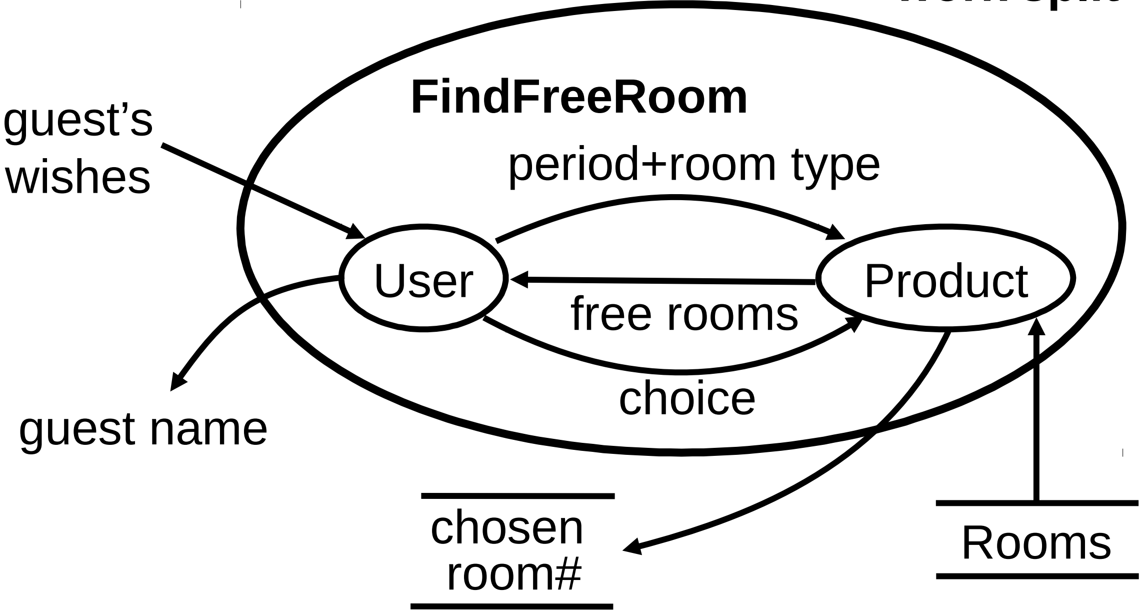
**The context is critical to how you do requirements engineering!**

# Fig 3.1    Human-computer - who does what?

**Domain model: parties joined**



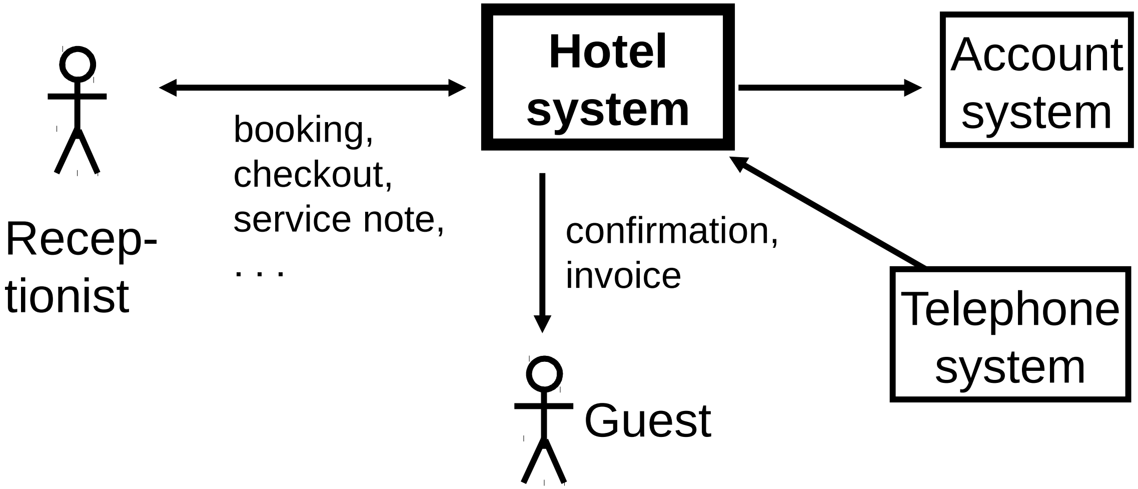
**Physical model: work split**



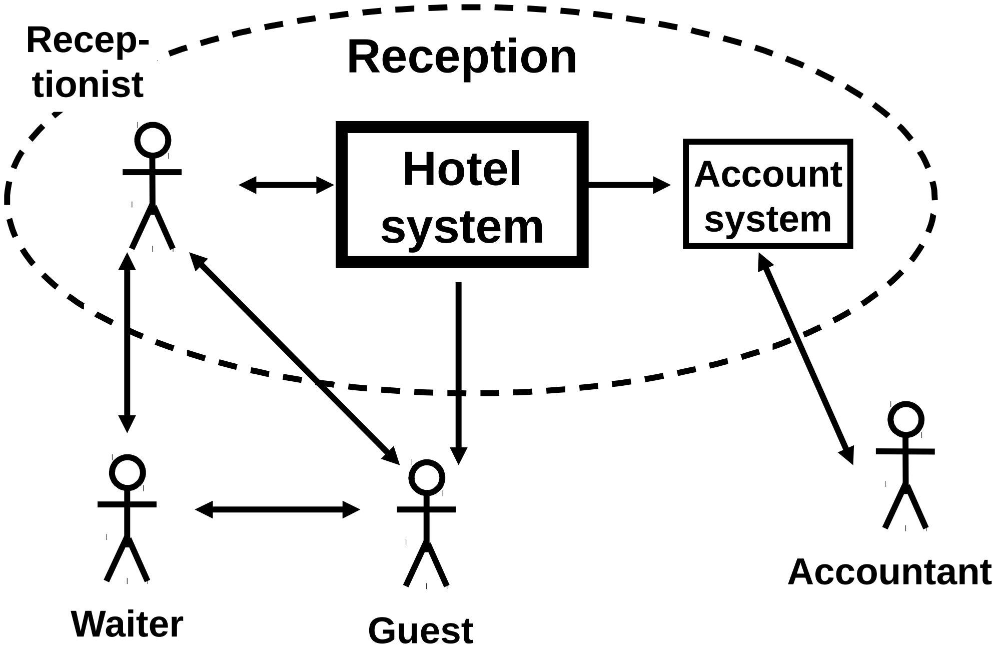
From: Soren Lauesen: Software Requirements

# Fig 3.2    Context diagram

**R1:**
The product shall have the following interfaces:



**R2 ??:**
The reception domain communicates with the surroundings in this way:



From: Soren Lauesen: Software Requirements

KOCKUMS

# The goal-design scale:
# Goal → Domain → Product → Design

Why?

Who?

What?

How?

```
Model(
  Goal("accuracy") has
    Spec("Our pre-calculations
          shall hit within 5%"),
  Feature("quotation") has
    Spec("Product shall support
          cost recording and
          quotation with
          experience data"),
  Function("experienceData") has
    Spec("Product shall have
          recording and
          retrieval functions
          for experience data"),
  Design("screenX") has
    Spec("System shall have
          screen pictures as
          shown in Fig. X")
)
```

***Mål-nivå*** bakomliggande syfte, affärsmål, användarnytta, effekt, vinst

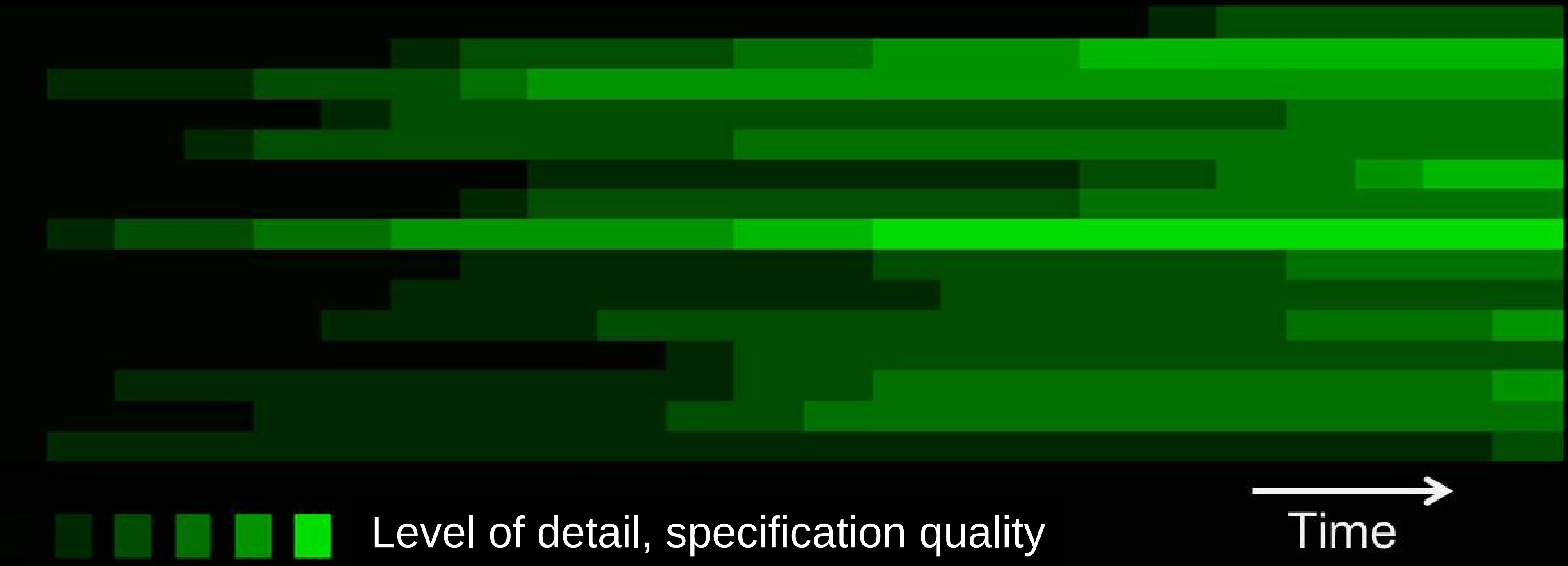
***Domän-nivå*** sammanhang, omgivning, hur användarna och produkten ***samverkar*** för att ge nytta

***Produkt-nivå*** externt observerbara funktioner och egenskaper

***Design-nivå*** specifik utformning av produktens innehåll

**Which level is best?**

# Evolving mix of levels of detail & quality in continuous requirements engineering



# Elicitation: [Lau:8]
# Get out there and dig up reqts!

”You *cannot* sit in your office and produce requirements based on intuition and logic.

You have to **discover** the non-trivial requirements from users and other stakeholders.”

[Lausen, page 42]

⇒ **PROVOKING AN UNDERSTANDING**

# Fig 1.6B Ask “why”

## Neural diagnostics

**System shall have mini keyboard with start/stop button, . . .**
**Why?**

**Possible to operate it with “left hand”.**
**Why?**

**Both hands must be at the patient.**
**Why?**

**Electrodes, bandages, painful . . .**

From: Soren Lauesen: Software Requirements

# Deep domain knowledge is critical to successful requirements engineering!!

# Why+How+Example

**Feature** navigate **has**

- **Why** Measuring neural response is a bit painful to the patient. Electrodes must be kept in place ... So both hands should be at the patient during a measurement.
- **Spec** It shall be possible to perform the commands start, stop, ... with both hands at the patient.
- **Example** Might be done with mini keyboard (wrist keys), foot pedal, voice recognition, etc.

See [Lausen] fig 1.6C

# Discussion Elicitation

- Why is elicitation so challenging in real projects?

# Fig 8.1 Elicitation issues

**Should be simple.**
**Ask stakeholders what they need!**

**Barriers:**

Cannot express what they need
Cannot explain what they do and why
May ask for specific solutions

Lack of imagination - new ways
Lack of imagination - consequences

Conflicting demands
Resistance to change

Luxury demands
New demands once others are met

**Things to elicit - intermediate work products:**

Present work, Present problems
Goals and critical issues
Future system ideas
Realistic possibilities

Consequences and risks
Commitment, Conflict resolution
Requirements, Priorities
Completeness

From: Soren Lauesen: Software Requirements

# Fig 8.2 Elicitation techniques

| | Present work<br>Present problems<br>Goals & key issues | Future system ideas<br>Realistic possibilities<br>Consequences & | Commitment<br>Conflict resolution | Requirements<br>Priorities<br>Completeness |
|---|---|---|---|---|
| Stakeholder analysis<br>(Group) interview<br>Observation<br>Task demo<br>Document studies<br>Questionnaires | | | | |
| Brainstorm<br>Focus groups<br>Domain workshops<br>Design workshops | | | | |
| Prototyping<br>Pilot experiments | | | | |
| Similar companies<br>Ask suppliers | | | | |
| Negociation<br>Risk analysis<br>Cost / benefit<br>Goal-domain analysis<br>Domain-reqs analysis | | | | |

**Nuläge**
**Framtid**
**Åtagande&samsyn**
**Krav & deras värde**

**Studier av & med (enskilda) intressenter eller dokument**

**Förberedda gruppaktiviteter**

**Exekvering av system**

**Omvärldsanalys**

**Avvägningar, risker, analys av kopplingar mellan nivåer**

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

## **Stakeholder analysis** (intressentanalys)

Example: In-house (internprojekt)

- Sponsors – want value for their money
- Users at different departments
- Managers at different departments
- Authorities, security managers, accountants etc.
- System management and support,
- Other indirect stakeholders that may provide valuable input

Example: product development:

- Distribution channels and retailers
- Solution providers building on your product
- Competitors

# Interviews in requirements elicitation

On or more stakeholders are interviewed by a requirements engineer (aka analyst)

Probably the most common elicitation method.

Reflect on pros and cons with ...

- Individual or group interviews?
- **Structured**: prepared questions and perhaps also response alternatives
- **Semi-structured**: some Q prepared, but freedom in order and depth
- **Unstructured**: no preconceived closed questions; open questions to start off: “What is your view on the system?”

# Intervjuer för att elicitera krav

En eller flera intressenter tillfrågas av en kravingenjör.

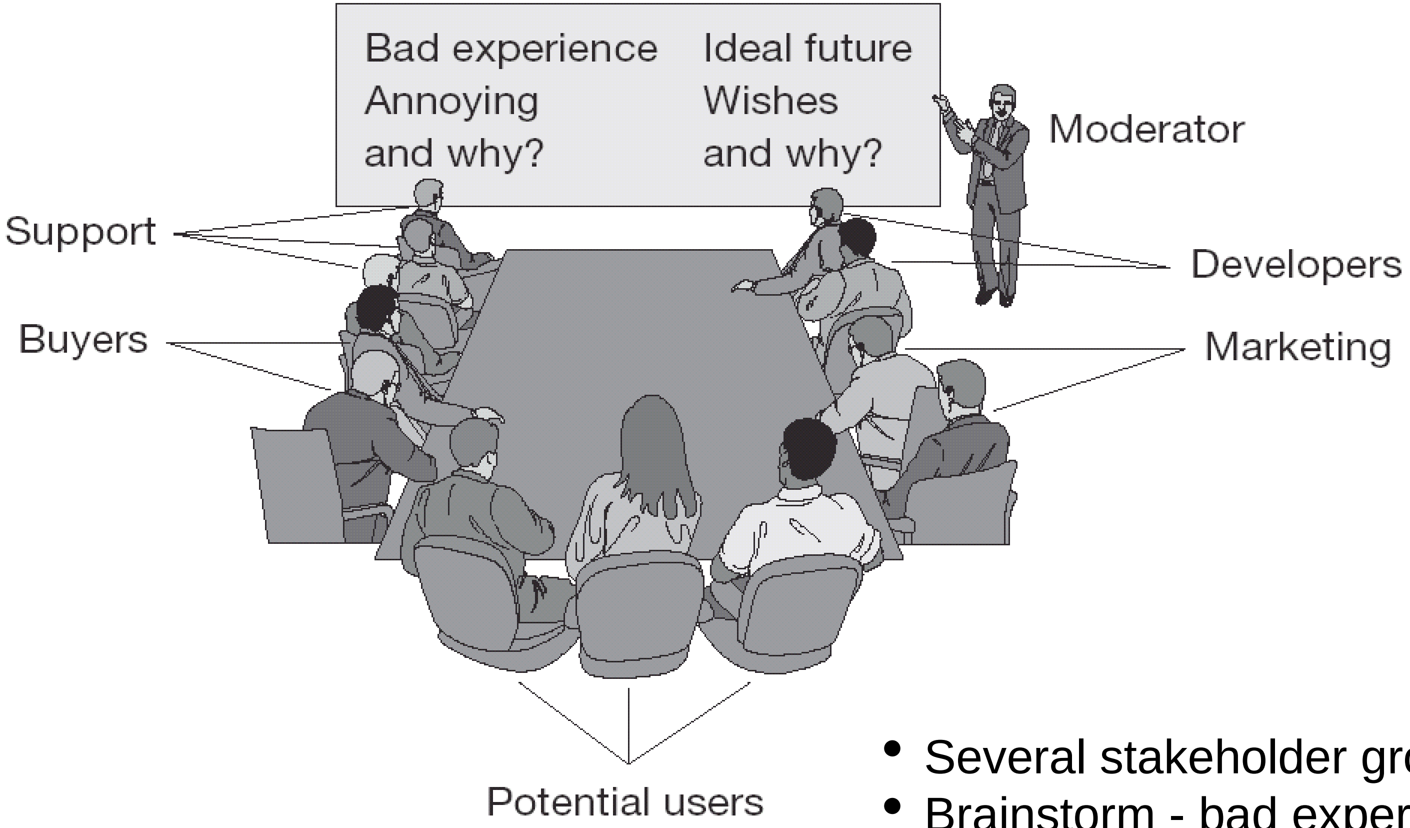
Förmodligen den vanligaste metoden.

För och nackdelar med…

- Enskilda eller gruppvisa intervjuer?
- **Strukturerade**: förbestämda frågor, ev förbestämda svarsalternativ
- **Semi-strukturerade**: vissa frågor är förberett men frihet i ordning och djup
- **Ostrukturerade**: inga förberedda frågor alt. några få öppna frågor

  ”Berätta om din syn på systemet?”

# Fig 8.4 Focus groups



- Several stakeholder groups
- Brainstorm - bad experience
- Brainstorm - wishes & ideal future
- Each group selects top ten issues
- A few days later: Decide.
- Each group must get something

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

Concrete example in Lau:10.2

# Requirements Prioritization

# Book chapter [PRIO]

”Requirements Prioritization”,  
Patrik Berander and Anneliese Andrews,  
*Engineering and Managing Software Requirements*,  
Eds. A. Aurum and C. Wohlin, Springer,  
ISBN 3-540-25043-3, 2005

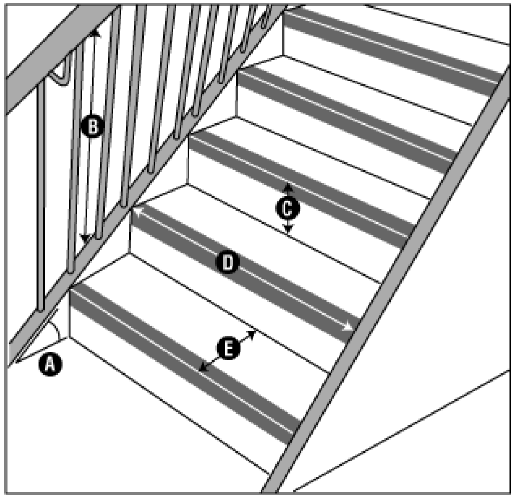
# Why prioritize?

- To focus on the most important issues
- To find high and low priority requirements
- To implement requirements in a good order
- To save time and money

[PRIO]

# Steps we need to do...

- Select prioritization apects
- Select prioritization objects (e.g., features)
  - Example: Define features at high level that can be selected or de-selected independently
- Structure and grouping
- Do the actual prioritization
  - Decide priorities for each aspect and object
- Visualize, discuss, iterate...



[PRIO]

# Prioritization challenges

- Finding a good abstraction level
- Combinatorial explosion
- Inter-dependencies
- Not easy to predict the future
- Power and politics

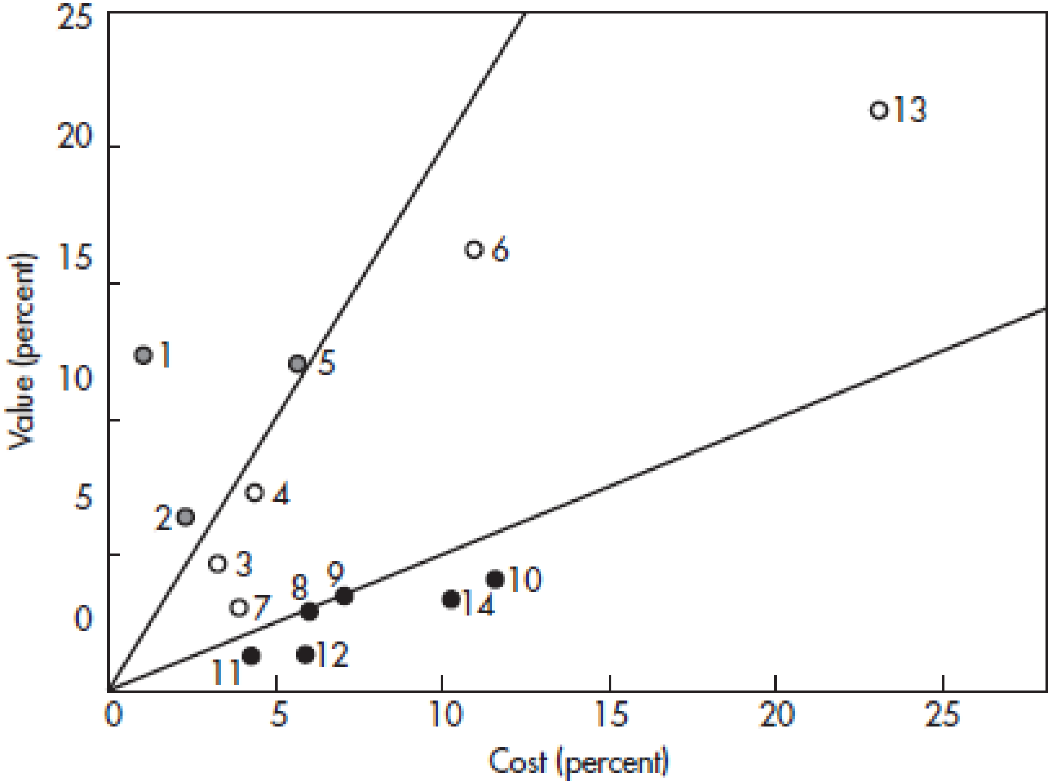
[PRIO]

# Prioritization aspects

- Example aspects:
  - Importance (e.g., financial benefit, urgency, strategic value, market share etc.
  - Penalty (if requirement not included)
  - Cost (e.g., staff effort)
  - Time (e.g., lead time)
  - Risk (e.g., technical, market)
  - Volatility (instability, risk of change)
- Other aspects (e.g., competitors, brand fitness, competence, release theme…)
- Combination of aspects: cost vs. benefit, cost vs. risk, importance vs. volatility
- Optimize: minimize or maximize some combinations, e.g., cost-benefit

[PRIO]

# Combining two criteria: Example Cost vs Benefit



Karlsson, Joachim, and Kevin Ryan. "A cost-value approach for prioritizing requirements." *IEEE software* 14.5 (1997): 67-74.

# When to prioritize?

- At decision points (toll gates), e.g.,
  - Project start
  - Start of construction
  - Release planning
  - Increment planning
- When big changes occur
- Iteratively with lagom intervals

[PRIO]

# Who should prioritize?

- Find the right competence for the right aspect
  - Developers know about e.g., development effort and engineering risk
  - Support organization knows about e.g., customer value and penalty
  - Marketing organization knows e.g., about competitors
  - etc...

[PRIO]

# Typical industry praxis

- Numerical assignment, e.g., 1-5
  - See Lauesen, chapter 7.4
- Problem:
  - Requirements are not confronted to each other
  - What should go out when new requirements wants in?
  - Everything tends to be important

    → decision avoidance

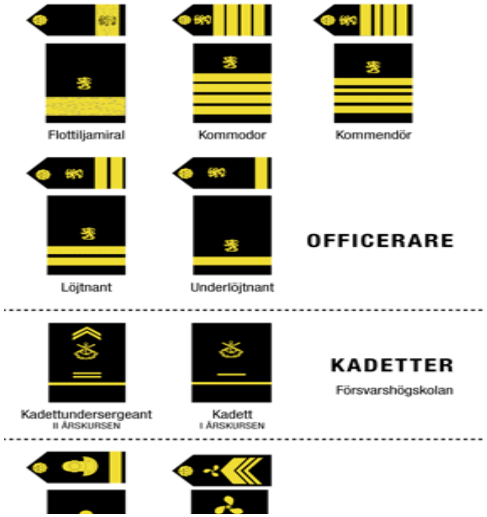
[PRIO]

# Prioritization scales

## Categorization

e.g.: must, ambiguous, volatile

Partition in groups without greater-less relations



## Ordinal scale

e.g.: more expensive, higher risk, higher value

Ranked list
A>B

## Ratio scale

ex: $, h,

% (relative)

Numeric relations:
A=2*B

[PRIO]

# Prioritization techniques

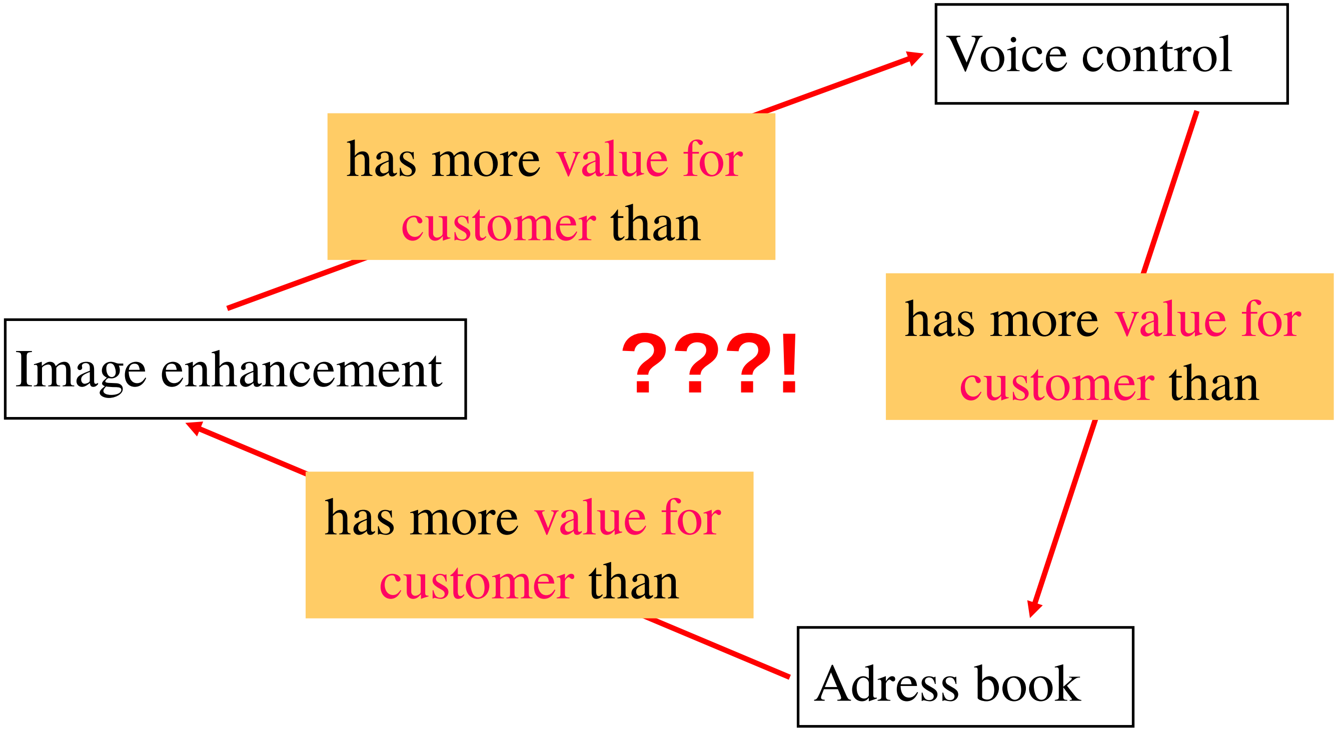
- **Grouping, Numerical assignment (grading)**
  - Can be done using any scale (categorical, ordinal, ratio)
  - Quick & easy; but a risk is that all reqs are deemed highly important as they are not challenged against each other; may be misinterpreted as ratio scale (even if ”4” not necessarily is "twice as much" as ”2” when using an ordinal scale).
- **100$-test**
  - Ratio scale, quick and easy, risk of shrewd tactics (listigt taktikspel)
- **Ranking (sorting)**
  - Ordinal scale, pairwise comparison, easy and rather quick
- **Top-ten (or Top-n)**
  - Ordinal scale if the top list is ranked / Categorical grouping if not ranked; very quick and simple, gives a rough estimate on a limited set of req
- (Analytical Hierarchy Process (AHP))
  - Ratio scale, pairwise comparison, tool is needed, redundancy gives measure of consistency

[PRIO]

# Combination of prioritization techniques

- Example of how prioritizaion techniques can be combined:
  1) Do a high-level **grouping**,
  2) Then **sorting** or **top-5 depending** of which group
  3) If a group is large: define sub-groups and use 100$-method

# Tools can help find inconsistencies



Voice control

has more value for customer than

Adress book

has more value for customer than

Image enhancement

has more value for customer than

???!

This will be illustrated at Lab 1.

[PRIO]

# Course ombudsmen – volunteers?

- A course ombudsman collects experiences and improvement suggestions during the course and gives this feedback to the teachers
- Any course member are encouraged to talk to the teachers about how the course works!
- When the Course Experience Questionnaire (CEQ) is ready, the two ombudsmen meets with the course head and discusses the working report and writes a short summary.
- D och eller C:are och någon/några från annat program?
    - TBD

http://www.dsek.lth.se/sektionen/srd/kursombud/

# To Do

- Meet with startup As Soon As Possible (ASAP!!!)
- Read Lau:1, 3.2, 8,  PRIO
- Go to exercise!
- Sign up for labs: link on course homepage
- Discuss in each project your ambitions and how you would like to work, solve conflicts, etc.
- Meet with project supervisor W2
- Hand in PM v2 on Thursday W2