

# Seven cost-efficient ways to reduce your project overruns

Kjetil Moløyken-Østfold – Conceptos Consulting

JavaZone 2010, 8th-9th September, Oslo

# About the speaker:

## Kjetil Moløkken-Østvold

- Senior Partner at Conceptos Consulting
- Academic and research background
  - PostDoc and Assistant Director at Simula Research Laboratory (Norway)
  - MSc (Siv.Ing.) and PhD from the University of Oslo (Norway)
  - Published 23 papers on various topics, including estimation, project management, agile, collaboration, communication, contracts etc.
- Recent commercial projects
  - External quality assurance and consulting for several large public sector projects (>500 MNOK)
  - Process improvement for financial and telecom companies



# Contents

- Introduction
- Challenges and recommended practices
- Summary
- Q&A

# Software Project Overruns

- About 70-80% of all projects encounter effort (cost) overruns<sup>1</sup>
- The average magnitude of effort overruns is 30-40%
- Similar results for schedule overruns
- No apparent change the past 30-40 years

<sup>1</sup> Moløkken-Østvold, Jørgensen, Tanilkan, Gallis, Lien and Hove. *A Survey on Software Estimation in the Norwegian Industry*, In 10th International Software Metrics Symposium (METRICS 2004)

# Agile development and estimation?

*”-the concept of an overrun is not one typically found in agile development processes themselves, and precision estimation up front is not typically seen as a priority.”*

- Comment from anonymous reviewer, Agile 2007, research track.

# Main challenge

- It appears as if many books, papers and tutorials in the Agile community assumes:
  - A customer with no need for budget or schedule when starting a project
  - No need for long term planning within the development team(s)
  - That minor estimates derived as you go (e.g. for sprints) are sufficient

# Estimating Agile projects

- Agile projects are hardly immune to overruns, delays and bad business decisions based on poor estimates
- “Planning Poker” or similar techniques for estimating sprints or releases are important, but not sufficient
- It is often necessary to provide a relatively accurate estimate of total project delivery schedule and costs at an early stage, due to:
  - Bidding
  - Budgeting
  - Staffing
  - Scheduling
  - Release planning
  - All of the above...

# Challenges and recommended practices

# Challenge #1: Ad-hoc estimation process

## Problems

- No agreed upon practices for estimation
- No documentation of estimates and experiences
- No learning from experience

## Recommended practice

- Checklists and experience databases – Cost efficient and adaptable

# Checklists

- A checklist is a simple way of utilizing experience
- A checklist can help the estimators to avoid overlooking tasks, which has been cited as a major reason for estimation inaccuracy
- One study showed that the use of a checklist reduced the estimators' over-optimism and increased estimate transparency and consistency

# Example structure

- Generic checklist(1) downloadable from <http://bit.ly/dsGE23>
- Consisting of four phases and twelve activities

(1) M. Jørgensen, and K. J. Moløkken-Østvold (2003) , A Preliminary Checklist for Software Cost Management, IEEE International Conference on Quality Software, pp. 134-140, Dallas, USA,

# PREPARATION PHASE

1. Understand estimation problem
2. Agree on decisions and assumptions relevant for estimation
3. Collect information relevant for estimation
4. Select or design the estimation process

# ESTIMATION PHASE

5. Estimate most likely cost
6. Assess uncertainty of cost estimate
7. Review estimation process

# APPLICATION PHASE

8. Apply estimate in bidding
9. Apply estimate in planning and budgeting
10. Communicate estimate (including communication of uncertainty, bid, plan and budget)
11. Control cost (including re-estimation)

# LEARNING PHASE

## 12. Learn from estimation feedback

# Use of experience data

- The idea of recording and utilizing data from experience is widely utilized in the software industry
- One of the strengths of this approach is that estimates are based on actual experience
- Findings regarding its utility are mixed
  - The problem is the often very unique nature of software development projects
  - Must be tailored to organization
- Also useful in order to assure external parties (customers, quality advisors etc.)

# Challenge #2: Estimating for other purposes than realism

## Problems

- Political pressure
- Bidding to win
- Trying to project competence by providing low estimates

## Recommended practice

- Encourage realism – separate the estimation process from other processes, such as bidding

# Organizational issue

- Political estimates are common in both in-house development and contracting
- Ideally, there should be a complete separation of estimation and planning/bidding
- Difficult to manage in practice

# Challenge #3: Lack of collaboration between customer and contractor

## Problems

- Blame game
- Lawyers
- Wasted time and effort

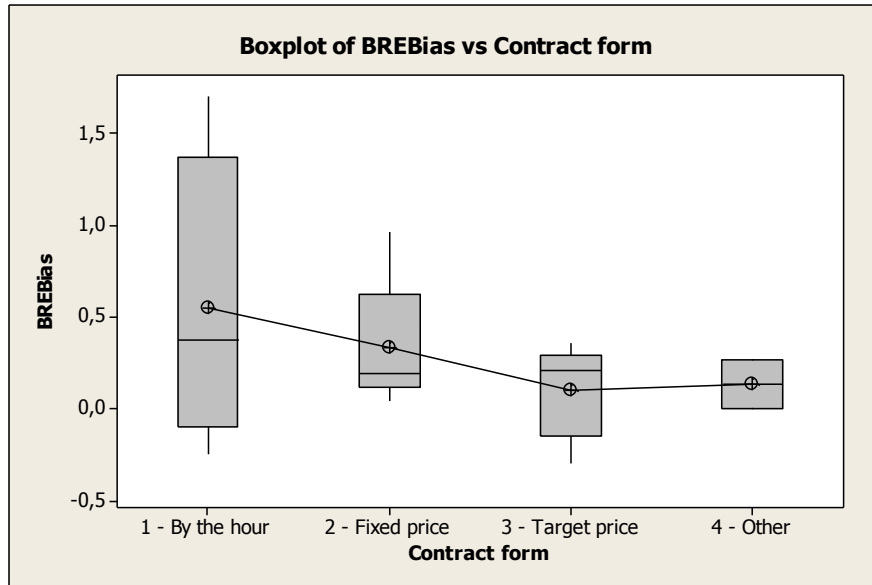
## Recommended practice

1. Target-price contracts (e.g. PS 2000)

# Contracts for collaboration

- Contracts are important since they often regulate collaboration (directly or indirectly)
- Common contract types
  - Time and material
  - Fixed price
  - Target price (PS2000)
    - Mutual sharing of cost overruns (and vice versa)
    - Floors and ceilings for cost sharing

# Contract form and overruns



	<i>Mean</i>	<i>Median</i>
By the hour	0.55	0.37
Fixed price	0.33	0.19
Target price	0.10	0.21
Other	0.13	0.13

# Example: PS2000

- Developed by the Norwegian Computer Society
  - Based on target price, collaboration facilitation and risk-sharing
  - Used by public and private sector projects, both large and small
  - Developed by experts representing customers, contractors and neutral advisors
- Adapted to agile processes in 2008-2009
  - Tutorial for PS2000 and agile released in 2008
  - Contract updated in new version in 2009
- Link: <http://bit.ly/9FN03L>

# Challenge #4: Management forcing estimates on project participants

## Problems

- Little or no input from developers
- Fixed scope/schedule for project/release
  - Before estimates for iterations are developed

## Recommended practice

- Collaborative estimation
  - Delphi, Wideband Delphi and other processes

# Research findings

- Research has found combination to increase estimation accuracy in several areas, including software development (1)
- The method chosen is not necessarily important
  - Different methods suit different scenarios
  - Often, combination in itself is more important than the method chosen

(1) K. J. Moløyken-Østfold, N. C. Haugen, and H. C. Benestad, Using Planning Poker for Combining Expert Estimates in Software Projects, Journal of Systems and Software 81(12):2106-2117(<http://bit.ly/9vb145>)

# Why combine estimates?

- Obtain knowledge from various sources
- Avoid extreme decisions
- Synchronize perceptions about estimates and work at hand
- Create ownership of estimates
- Remove irrelevant information (if using moderator)
- Introduce a "devils advocate"

# Pitfalls when combining estimates

- Passive participants
- Depending on chosen method: political pressure (groupthink)
- Requires good moderators and experts
- Time-consuming and costly (?)

# Some methods for combining estimates

<i>Method</i>	<i>Structure</i>	<i>Anonymity</i>	<i>Interaction</i>	<i>Overhead</i>
Delphi	Heavy	Yes	No	Major
Wideband Delphi	Moderate	Limited	Limited	Moderate
Planning Poker	Light	No	Yes	Limited
Unstructured groups	Light	No	Yes	Limited
Statistical groups	Light	Yes	No	Limited
Decision markets	Heavy	Yes	No	Moderate

# Delphi

- The Delphi technique was developed by RAND as a method for eliciting and refining group judgment<sup>1</sup>
- Three main features:
  - Anonymous responses (questionnaires)
  - Iterations and controlled feedback (several rounds, with moderator)
  - Statistical group responses (final round is verdict)

<sup>1</sup> [http://www.rand.org/pubs/research\\_memoranda/RM5888/](http://www.rand.org/pubs/research_memoranda/RM5888/)

# Wideband Delphi

- The technique is a modification of the Delphi technique, developed by Boehm and Farquhar
- Similar to the Nominal Group technique, also known as the estimate-talk-estimate technique
- The experts meet for group discussions both prior to, and during, the estimation iterations
- Inspiration for *Planning Poker*

# Challenge #5: Failure to adapt the estimation process to the magnitude of requirements

## Problems

- Ad-hoc requirement handling
- Process not adapted to size of project
- No clear definitions of roles and responsibilities
- "Static" requirements in word documents etc.
- Several tools/documents at once, used by various people and not updated

## Recommended practice

- Structured requirement handling – with integrated estimation

# Requirement handling

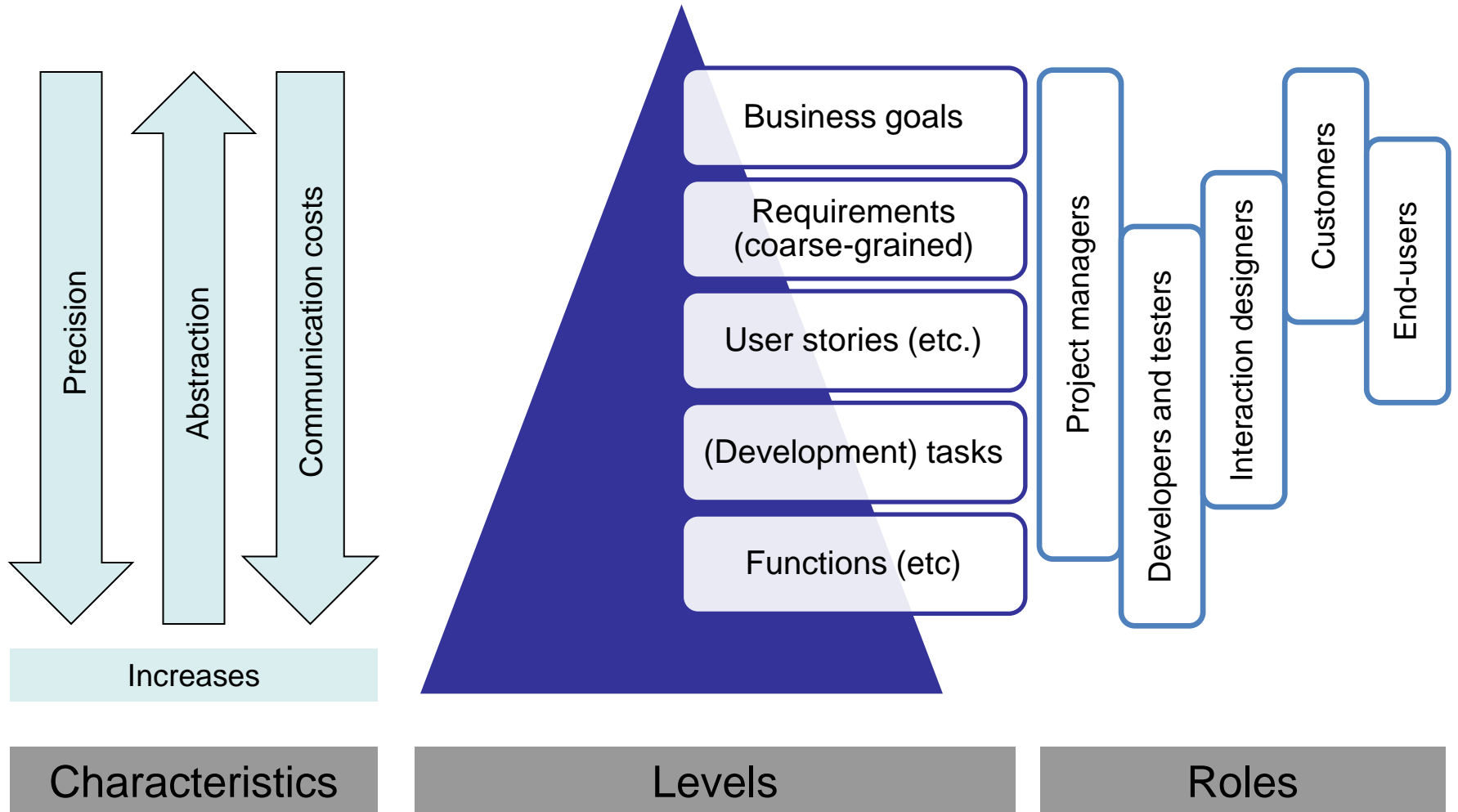
- Changed and new requirements are perceived as the customers' most frequent contribution to overruns<sup>1</sup>
- Overruns are prevented by the availability of competent customers and capable decision makers
- Avoid the influence of irrelevant and misleading information

<sup>1</sup> S. Grimstad, M. Jørgensen, and K. J. Moløkken-Østvold. The Clients' Impact on Effort Estimation Accuracy in Software Development Projects, In: 11th IEEE International Software Metrics Symposium (METRICS 2005), Como, Italy, September 19-22, pp. 3, IEEE, 0 ed.. 0, 2005.

# Is requirement handling important?

- Requirement processes demand
  - Structure and a long-term perspective
  - Close collaboration with customers/users
- Should you have
  - A big up-front requirement process,
  - Only one for each iteration, or
  - A combination?
- Many have trouble with requirement processes in (agile) software development

# Example visualization: the requirement pyramid



# Estimation should be an integrated part of the requirement process

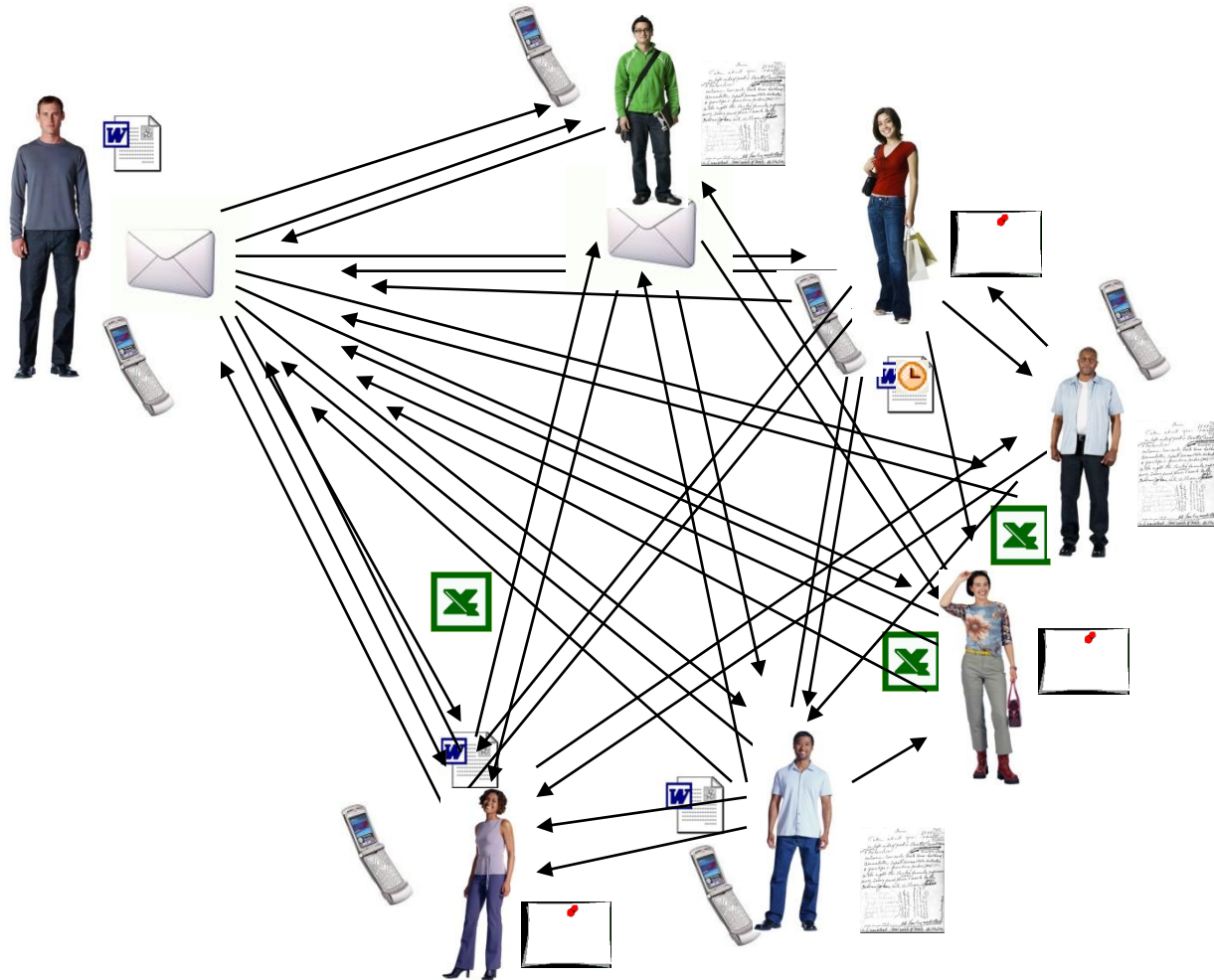
- Estimation, prioritization and planning of requirements provides new information
- Several roles should be involved in the estimation
  - Consider involving the customer as well
- Estimation should happen at several "levels" (ref. the pyramid) in order to verify and triangulate

**Describing, refining,  
prioritizing and  
communicating  
requirements is difficult  
for business people**



**Understanding, estimating  
and developing  
requirements is difficult  
for software developers**

# Problem: inefficient communication



# Lessons learned

- Analyze your situation regarding quality of requirements and magnitude of effort overruns
- Get rid of static documents (Word, Excel etc.) shared via email and informal communication (calls, notes on desk etc.)
- Implement a web-based “live” collaboration platform, able to handle requirements, estimates and prioritization
  - See for example a talk from NDC on “Technology Supported Requirement Handling and Estimation”:  
<http://bit.ly/9yXeF3>

# Challenge #6: Lack of traceability and clarity of estimates

## Problems

- Original estimates are not retained
- Conflicting sets of estimates (many of them!)
- Nobody has “the big picture”
- No updating after revisions

## Recommended practice

- Accountability - For managers, product owners, developers, product sponsors and other stakeholders

# Conflict example

- Original estimates kept in various excel-sheets
- Final bid (.pdf/word to customer) different from estimates
- Estimates not updated in excel
- After development starts, estimates are revised on project platform, and inconsistent with both bid and original estimates

# Solution

- Only have one "master repository" of information regarding estimates
- Should be linked to information regarding requirements, for example on a web-based collaboration platform
- Estimates must *not* only be available and/or understandable by one individual

# Challenge #7: Misunderstandings and wrong assumptions

## Problems

- Time wasted on developing wrong and unwanted functionality
- Lack of shared experience and motivation

## Recommended practice

- Frequent communication

# Frequent communication

- Previous studies have found communication to be important for project success
- Frequent communication can be used to prioritize features, set focus on bug-fixing or include more functionality (Beck and Fowler, *Planning Extreme Programming*, 2001)
- Motivated in part by Cockburn<sup>1</sup>, we explored the *frequency of communication between the contractor and the customer*

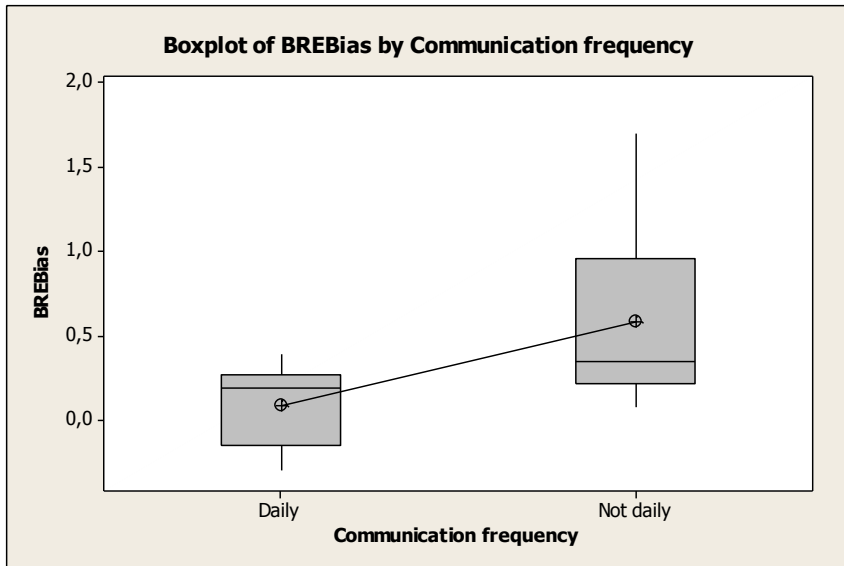
<sup>1</sup> Cockburn, "The End of Software Engineering and the Start of Economic-Cooperative Gaming," *ComSIS*, 2004.

# Frequent communication

- Previous studies have found communication to be important for project success
- Frequent communication can be used to prioritize features, set focus on bug-fixing or include more functionality (Beck and Fowler, *Planning Extreme Programming*, 2001)
- Motivated in part by Cockburn<sup>1</sup>, we explored the *frequency of communication between the contractor and the customer*

<sup>1</sup> Cockburn, "The End of Software Engineering and the Start of Economic-Cooperative Gaming," *ComSIS*, 2004.

# Contact frequency and overruns



<i>Level</i>	<i>Mean</i>	<i>Median</i>
Daily	0.09	0.19
Not Daily	0.58	0.35

- A Kruskal-Wallis test for difference results in  $p=0.023$
- The corresponding size of effect is  $d=1.25$ , indicating a large size of effect

# Summary

## Challenges

1. Using an ad-hoc estimation process
2. Estimating for other purposes than realism
3. Lack of collaboration between customer and contractor
4. Management forcing estimates on project participants
5. Failure to adapt the estimation process to the magnitude of requirements
6. Lack of traceability and clarity of estimates
7. Misunderstandings and wrong assumptions

## Recommended practices

1. Checklists and experience databases  
– Cost efficient and adaptable
2. Realism – Separate the estimation process from other processes, such as bidding
3. Target-price contracts (e.g. PS 2000)
4. Collaborative estimation – Delphi, Wideband Delphi and other processes
5. Structured requirement handling –  
With integrated estimation
6. Accountability - For managers, product owners, developers, product sponsors and other stakeholders
7. Frequent communication

# Contact information

- Presentation: <http://www.conceptos.no/>
- Email: [kjetil@conceptos.no](mailto:kjetil@conceptos.no)