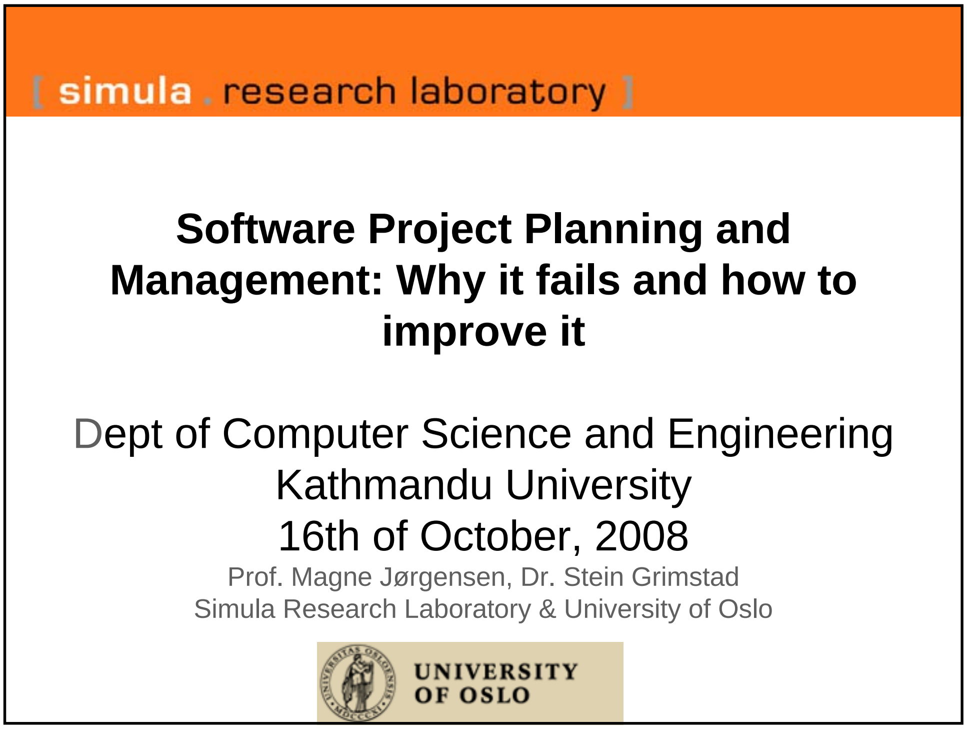


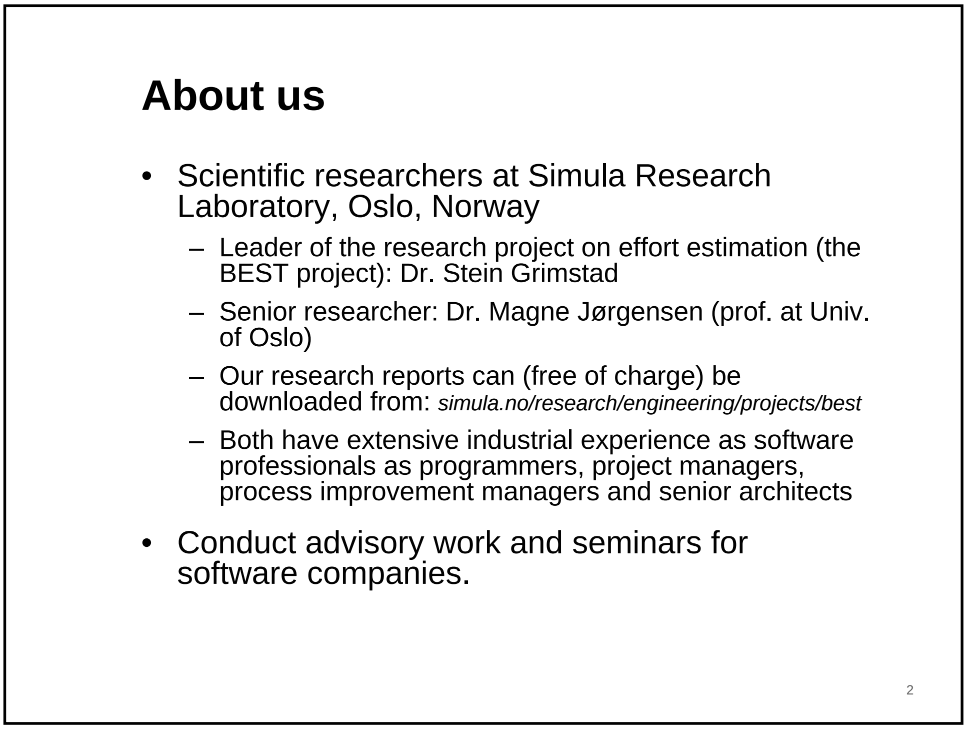
[simula . research laboratory]

Software Project Planning and Management: Why it fails and how to improve it

Dept of Computer Science and Engineering
Kathmandu University
16th of October, 2008

Prof. Magne Jørgensen, Dr. Stein Grimstad
Simula Research Laboratory & University of Oslo

UNIVERSITY OF OSLO



About us

- Scientific researchers at Simula Research Laboratory, Oslo, Norway
  - Leader of the research project on effort estimation (the BEST project): Dr. Stein Grimstad
  - Senior researcher: Dr. Magne Jørgensen (prof. at Univ. of Oslo)
  - Our research reports can (free of charge) be downloaded from: *simula.no/research/engineering/projects/best*
  - Both have extensive industrial experience as software professionals as programmers, project managers, process improvement managers and senior architects
- Conduct advisory work and seminars for software companies.

We are located in Norway ...

3

and like mountains and snow ...

4

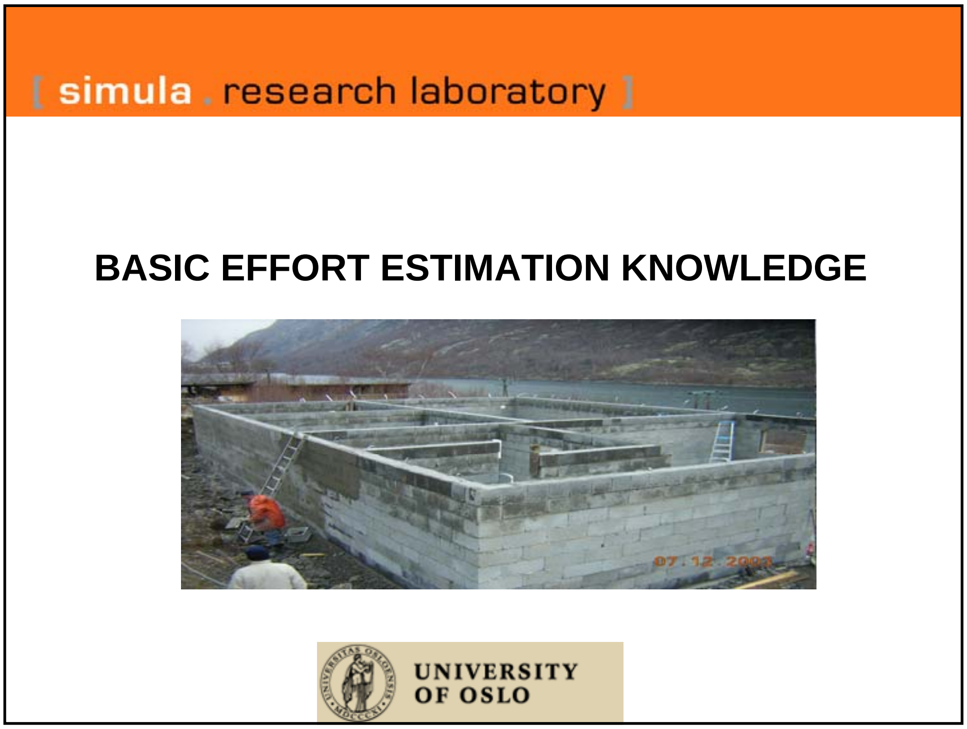
Agenda

10-12: Estimation of software development effort

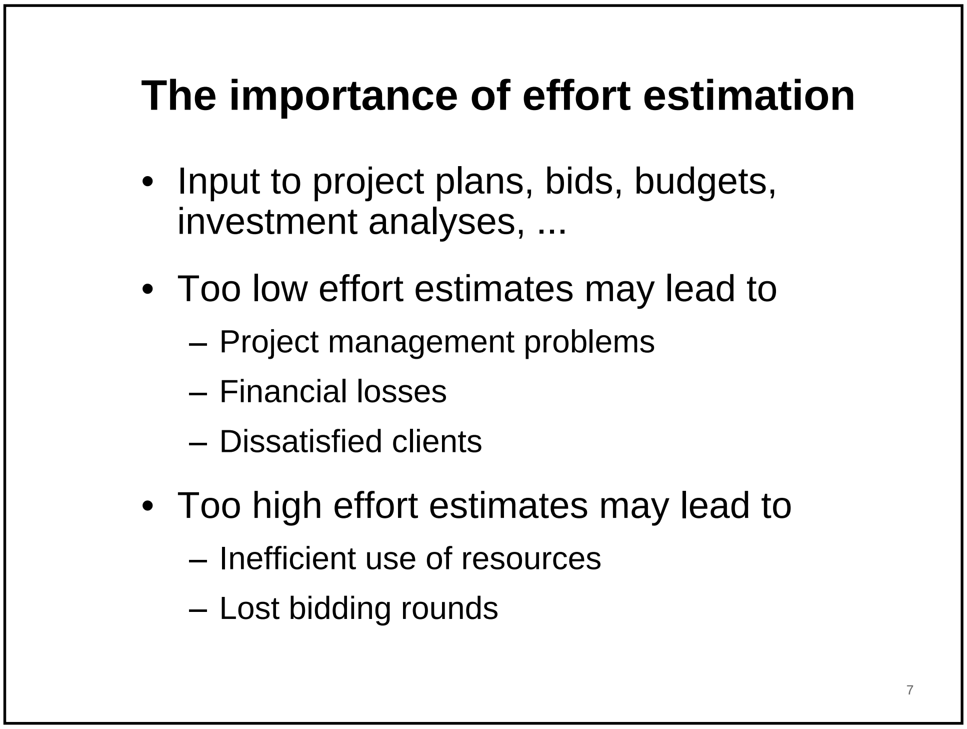
12-14: Pause (lunch)

14-15: Risk analysis and uncertainty management

15-16: Planning and management in agile projects

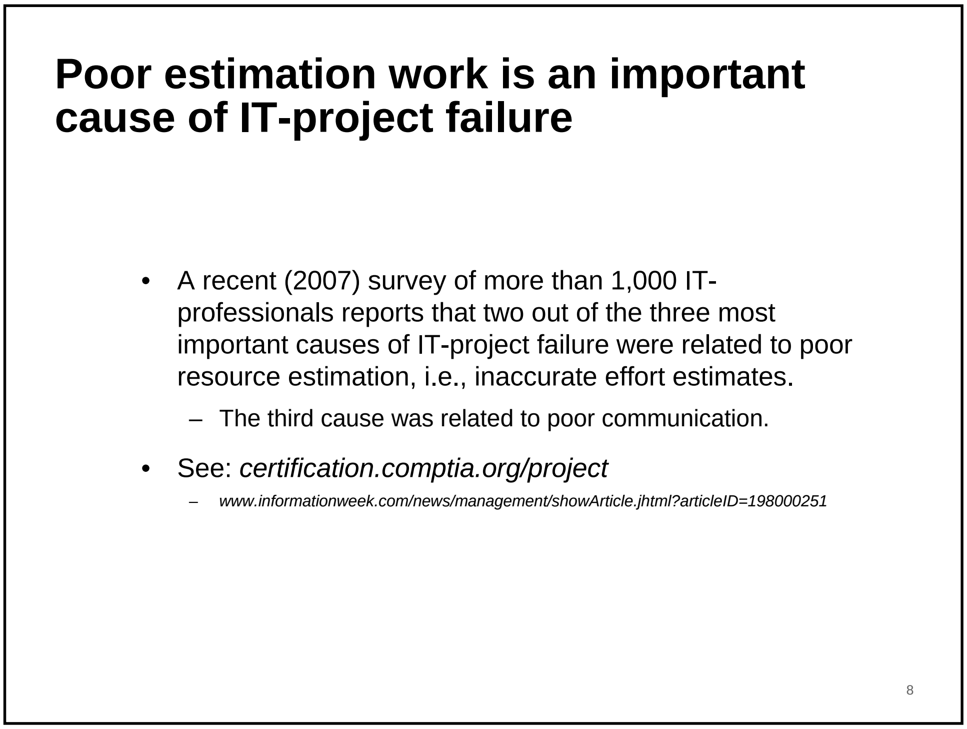


BASIC EFFORT ESTIMATION KNOWLEDGE



The importance of effort estimation

- Input to project plans, bids, budgets, investment analyses, ...
- Too low effort estimates may lead to
  - Project management problems
  - Financial losses
  - Dissatisfied clients
- Too high effort estimates may lead to
  - Inefficient use of resources
  - Lost bidding rounds



Poor estimation work is an important cause of IT-project failure

- A recent (2007) survey of more than 1,000 IT-professionals reports that two out of the three most important causes of IT-project failure were related to poor resource estimation, i.e., inaccurate effort estimates.
  - The third cause was related to poor communication.
- See: *certification.comptia.org/project*
  - *www.informationweek.com/news/management/showArticle.jhtml?articleID=198000251*

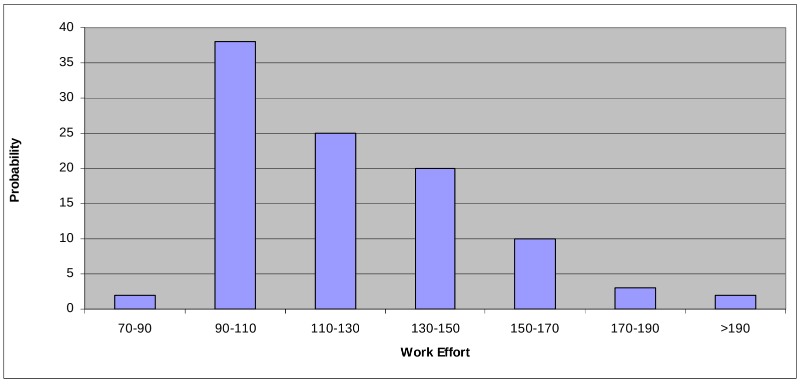
Estimation error

- Average estimation overrun in IT-projects is reported to be about 30%
  - Sometimes the estimation error is 200% and more.
  - Large estimation error sometimes causes huge project management, profitability, client satisfaction and investment analysis problems!
  - No substantial changes in average estimation error from 1970 until today. Why cannot we learn from previous experience?
- But first: What is the meaning of ”estimate” and ”overrun”?
  - www.simula.no/departments/engineering/publications/Grimstad.2006.1

What is an ”estimate”?

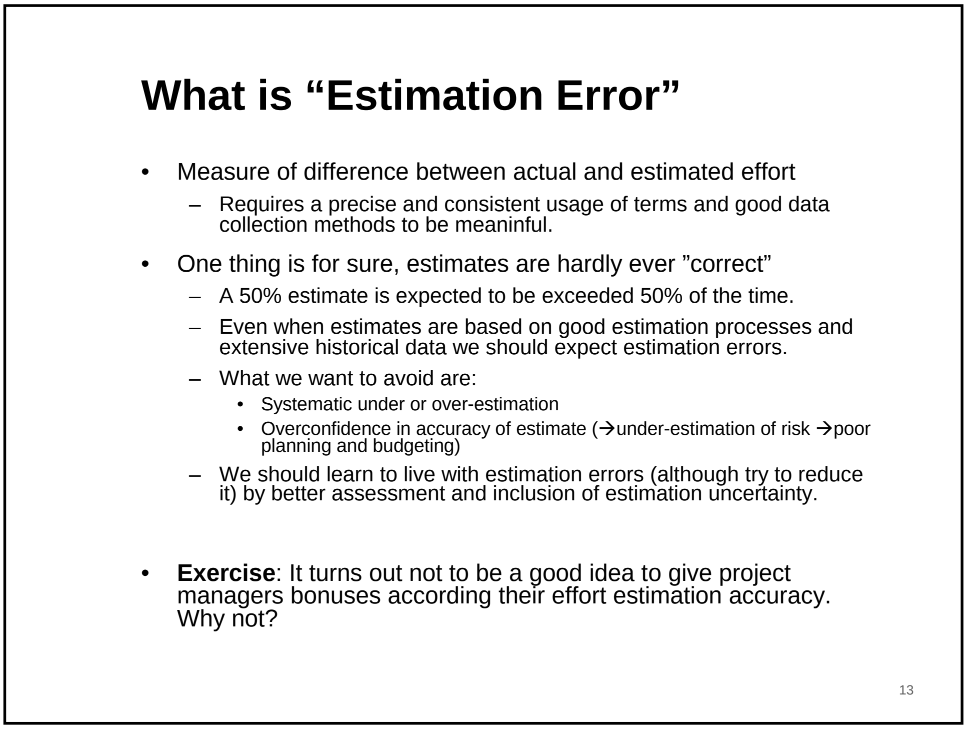
An effort estimate is the: i) most likely effort (mode), ii) 50% estimate (median), iii) most optimistic effort, iv) ideal effort, v) 70% estimate, vi) planned effort, vii) budgeted effort, viii) priced effort, ix) effort used as input to the bid, or, ...?

We have to think probabilistically about effort usage to enable good communication about what we mean by an effort estimate!



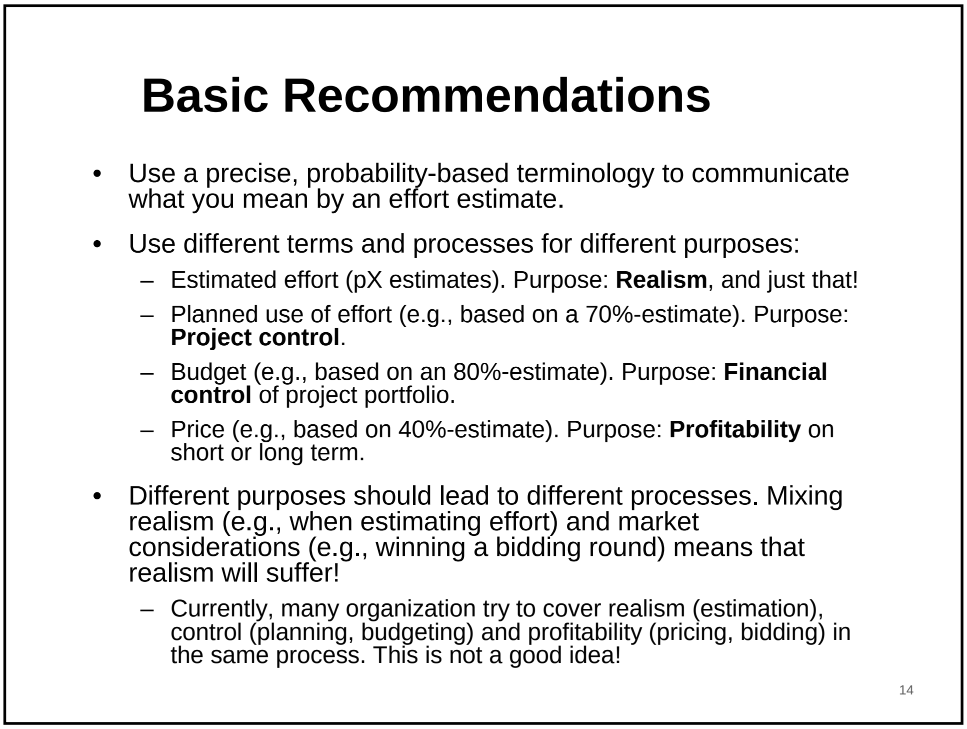
Recommendation: Use X% estimates

- Always inform about the type of estimate that you are providing (or receiving)
  - 50% estimate = just as likely to observe over- and under-run
  - 80% estimate = most likely effort + a risk buffer that makes it unlikely (only 20% likely) that there will be overruns. Could for example be the budget or the basis for the price to client.
  - 30% estimate = a close to best case estimate of the effort. Could for example be the bid in a situation where there are long term benefits of a client relationship.
- A method for the assessment of the likelihoods, (e.g., 80% likely not to exceed”) is presented later.



What is “Estimation Error”

- Measure of difference between actual and estimated effort
  - Requires a precise and consistent usage of terms and good data collection methods to be meaninful.
- One thing is for sure, estimates are hardly ever ”correct”
  - A 50% estimate is expected to be exceeded 50% of the time.
  - Even when estimates are based on good estimation processes and extensive historical data we should expect estimation errors.
  - What we want to avoid are:
    - Systematic under or over-estimation
    - Overconfidence in accuracy of estimate (→under-estimation of risk →poor planning and budgeting)
  - We should learn to live with estimation errors (although try to reduce it) by better assessment and inclusion of estimation uncertainty.
- **Exercise**: It turns out not to be a good idea to give project managers bonuses according their effort estimation accuracy. Why not?



Basic Recommendations

- Use a precise, probability-based terminology to communicate what you mean by an effort estimate.
- Use different terms and processes for different purposes:
  - Estimated effort (pX estimates). Purpose: **Realism**, and just that!
  - Planned use of effort (e.g., based on a 70%-estimate). Purpose: **Project control**.
  - Budget (e.g., based on an 80%-estimate). Purpose: **Financial control** of project portfolio.
  - Price (e.g., based on 40%-estimate). Purpose: **Profitability** on short or long term.
- Different purposes should lead to different processes. Mixing realism (e.g., when estimating effort) and market considerations (e.g., winning a bidding round) means that realism will suffer!
  - Currently, many organization try to cover realism (estimation), control (planning, budgeting) and profitability (pricing, bidding) in the same process. This is not a good idea!

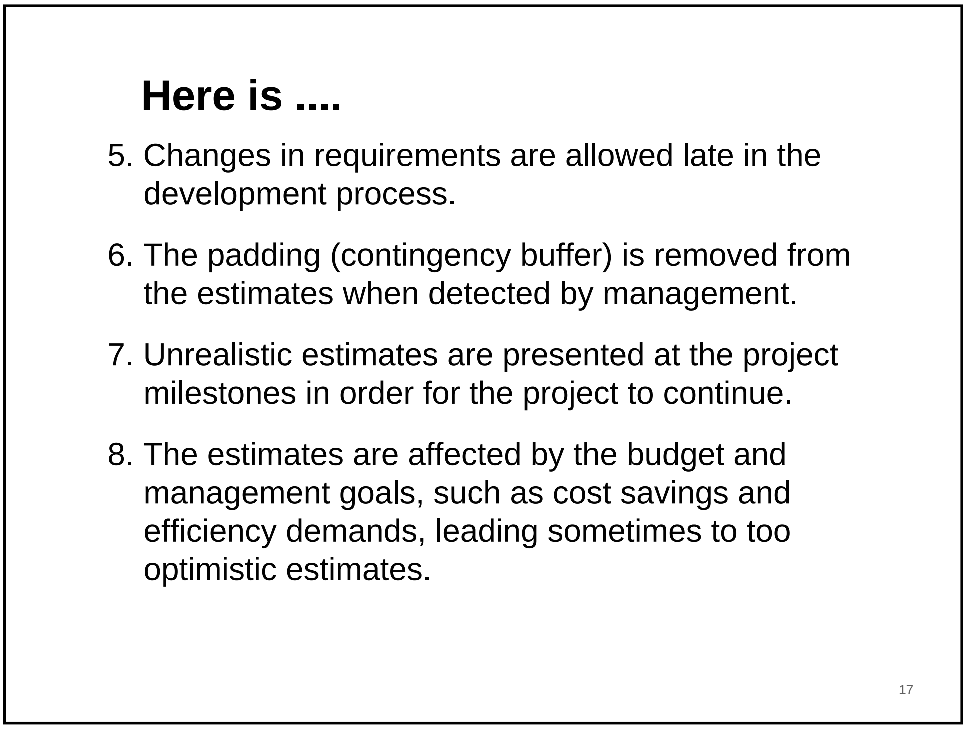
[simula . research laboratory]

Reasons for Estimation Error (and consequently IT project failure)

UNIVERSITY OF OSLO

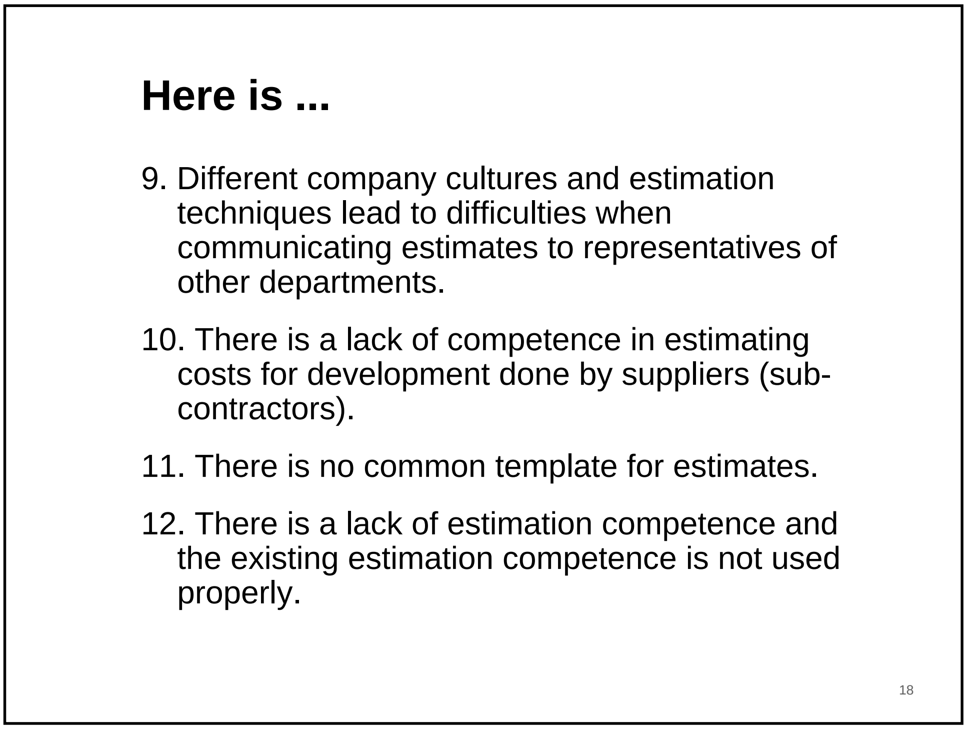
Here is one organization’s error reason list ...

1. Too little time spent on estimation work.
2. Management estimation goals are not taken seriously.
3. Dependences of other projects are not taken into account when estimating costs. (The staff resources might be moved to other, more critical projects to meet the deadlines.)
4. Unclear requirements make it difficult to know what is to be developed and estimated. (Especially producing early estimates for new technology is found to be difficult.)



Here is ....

5. Changes in requirements are allowed late in the development process.
6. The padding (contingency buffer) is removed from the estimates when detected by management.
7. Unrealistic estimates are presented at the project milestones in order for the project to continue.
8. The estimates are affected by the budget and management goals, such as cost savings and efficiency demands, leading sometimes to too optimistic estimates.



Here is ...

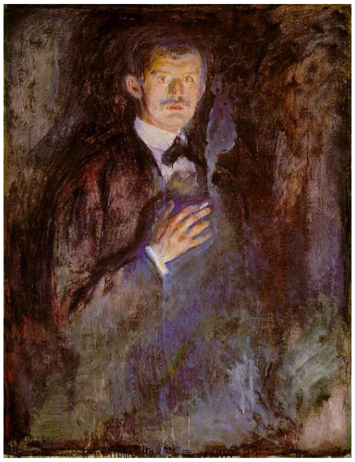
9. Different company cultures and estimation techniques lead to difficulties when communicating estimates to representatives of other departments.
10. There is a lack of competence in estimating costs for development done by suppliers (sub-contractors).
11. There is no common template for estimates.
12. There is a lack of estimation competence and the existing estimation competence is not used properly.

Can we trust that we know the real error reasons?

- “Reasons for Software Effort Estimation Error: Impact of Respondent Role, Information Collection Approach and Data Analysis Method”, published in *IEEE Transactions on Software Engineering*
  - *simula.no/research/engineering/publications/SE.4.Joergensen.2004.h*
- Main finding is that software developers are not very good at analyzing reasons for estimation error:
  - Direct reasons are typically emphasized and indirect reasons are forgotten
  - ”Systemic” reasons are seldom analyzed
  - Ability to know when patterns are random and when there is a cause-effect is low
- Reasons that are emphasized by software professionals depend very much on how you ask, whether the project has been a success or not, and how you analyze.
- **Recommendation**: Analyze reasons for overruns, but do not forget to analyze reasons for accurate estimates:
  - Be realistic about what you can learn about reasons for overruns
  - Ask ”why” five times
  - Apply ”Root cause analysis”

There are deeper explanations of the strong tendency towards over-optimism, such as ...

Slightly depressive people are those most realistic about their own performance



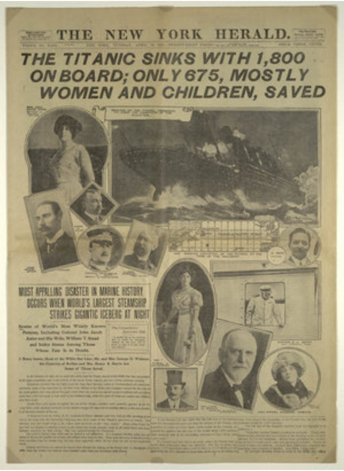
21

Overconfidence (under-estimation of risk) is normal, and the society benefits from this ...

22

We need optimists – but not necessarily as estimators

Over-confidence: This boat cannot sink ....



THE NEW YORK HERALD.

THE TITANIC SINKS WITH 1,800 ON BOARD; ONLY 675, MOSTLY WOMEN AND CHILDREN, SAVED

MOST APPALLING DISASTER IN MARINE HISTORY OCCURS WHEN WORLD'S LARGEST STEAMSHIP STRIKES GIGANTIC ICEBERG AT NIGHT

Categories of reasons for the strong tendency towards over-optimism

- Evolution/biology
- Culture
- Motivation
- Cognition
- Bidding round process

Biology

- Evolution rewards over-confidence and over-optimism

- Some (weak) indications that optimism improves coping ability.

- There are contexts where over-optimism is a more optimal strategy than realism. Particularly, when you know little about the probability of different outcomes, and more about the consequences of them.

Culture

- There may be cultural differences in level of optimism, but the findings are confusing and not consistent.
- Some studies suggest that those in collectivistic cultures are more over-optimistic, and other show the opposite result.
- There may be differences in how people from different cultures think when they estimate effort, but people seem to be over-optimistic in all cultures. We plan to study this.

Motivation

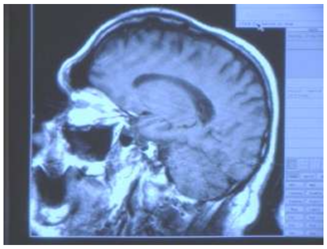
- Strong connection between high motivation for low use of effort and over-optimism
- Optimism can have a positive impact on performance, BUT
  - Only for a short period of time.
  - It’s easy to over-evaluate this.

Motivation (cognitive dissonance)

- A good self-evaluation is beneficial
  - For yourself
  - Because it’s used an performance indicator by others
- Low effort estimates = high performance = better (but less realistic) self-evaluation.
  - Otherwise, we have a cognitive dissonance, i.e., a difference between what we estimate and who we want to be.

Cognitive processes

- Planning (scenarios of the future) makes us more optimistic than looking back (use of historical data).
- Illusion of control sometimes very strong
  - Perhaps the most important reason for over-optimism?
- Over-optimisms increases with “psychological distance”



Cognitive processes

- Selective memory
- Cause-effect analysis may lead to better understanding, but also to over-estimation of how much better the understanding really is
- Hindsight bias

Bidding round format frequently leads to over-optimism

- The winner’s curse
- Bidding anchors
- Wishful thinking (future opportunities)

Recommendations to reduce over-optimism

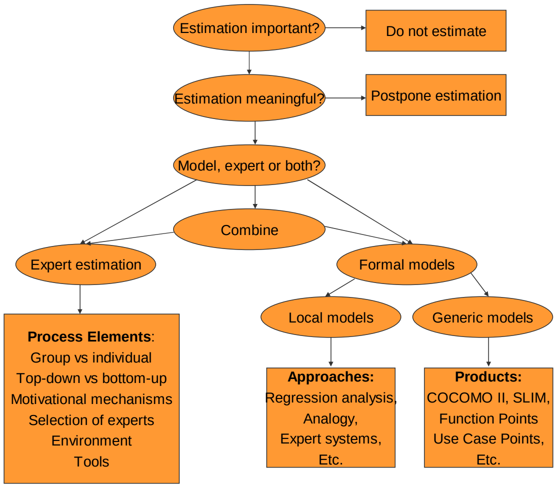
1. Educate a ”cost engineer” that will be evaluated wrt realism of estimates and not him/herself be a part of the projects estimated.
2. Use separate processes (and people?) for estimation, planning and bidding.
3. Avoid irrelevant information (prepare information material before given to the estimators)
4. Use historical data
5. Ask for estimation justification based on historical data. Require very good arguments if the estimates are based on assumption of much less effort compared to similar projects.
6. Do not assume that you have learned very much from previous projects.
7. When there are no relevant historical data available, try to find experts with relevant experience and historical data outside the organizations.
8. Do not let the most skilled estimators estimate the effort of junior developers. Use instead medium skilled developers.
9. If a person benefits from low effort estimates (really wants to start the project etc.), find another person to estimate the effort.
10. Combine estimates from different sources. Use a Delphi-like process (e.g., Planning Poker) to combine these estimates.

[simula . research laboratory]

Selection of Estimation Method

UNIVERSITY
OF OSLO

A Decision Tree for Method Selection

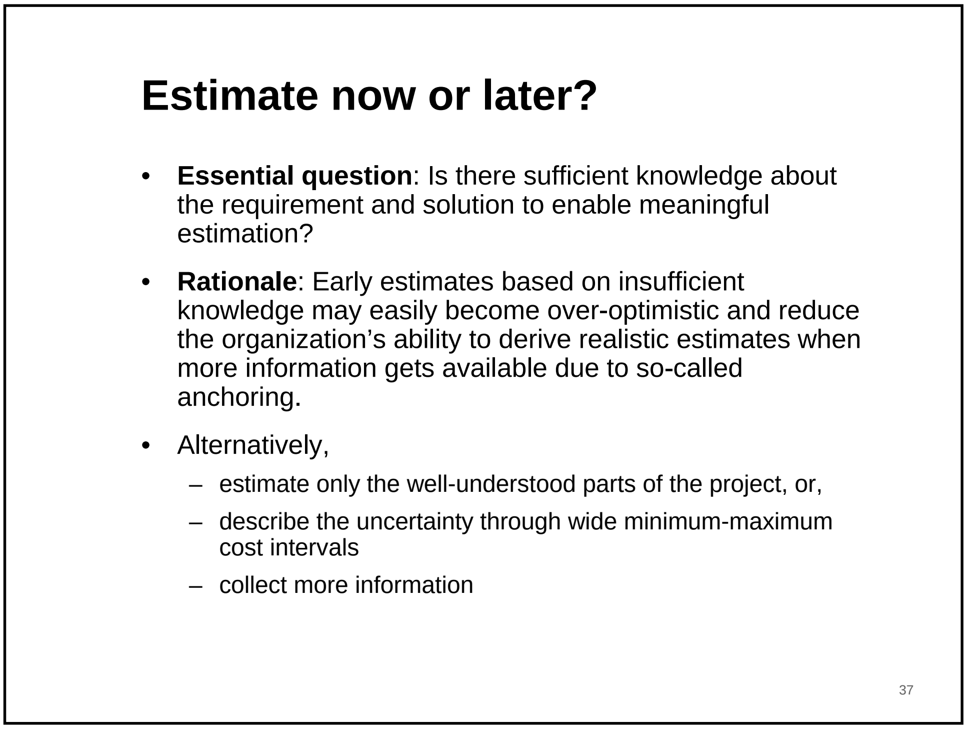


[simula . research laboratory]

15

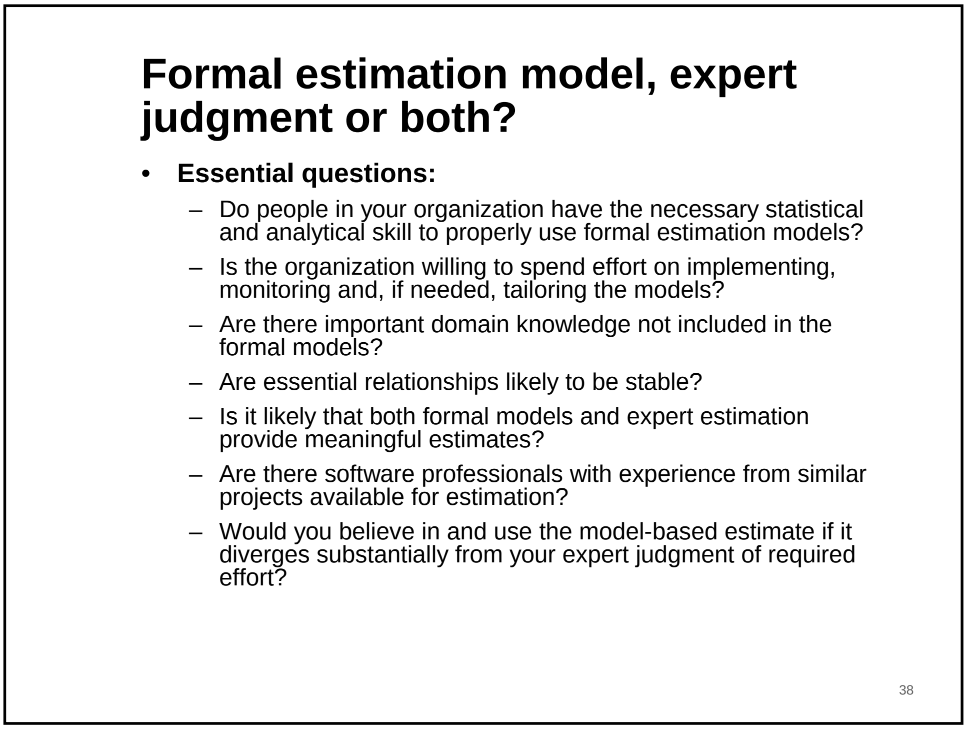
Estimate or not estimate?

- **Essential question**: Do you really need a cost estimate?
- **Rationale**: An estimate, if it is too low or too high, frequently have unwanted impacts on the project behavior, e.g., poor design (too low estimate) and “gold plating” (too high estimate).
- There are several alternatives to estimation that should be considered, such as:
  - Incremental development with the philosophy of do as much as possible within budget, starting with “need to have”-functionality.
  - The client has selected you on the basis of the belief (and previous history in support of this) that the company will work efficiently and with proper quality and says “Just do it!”



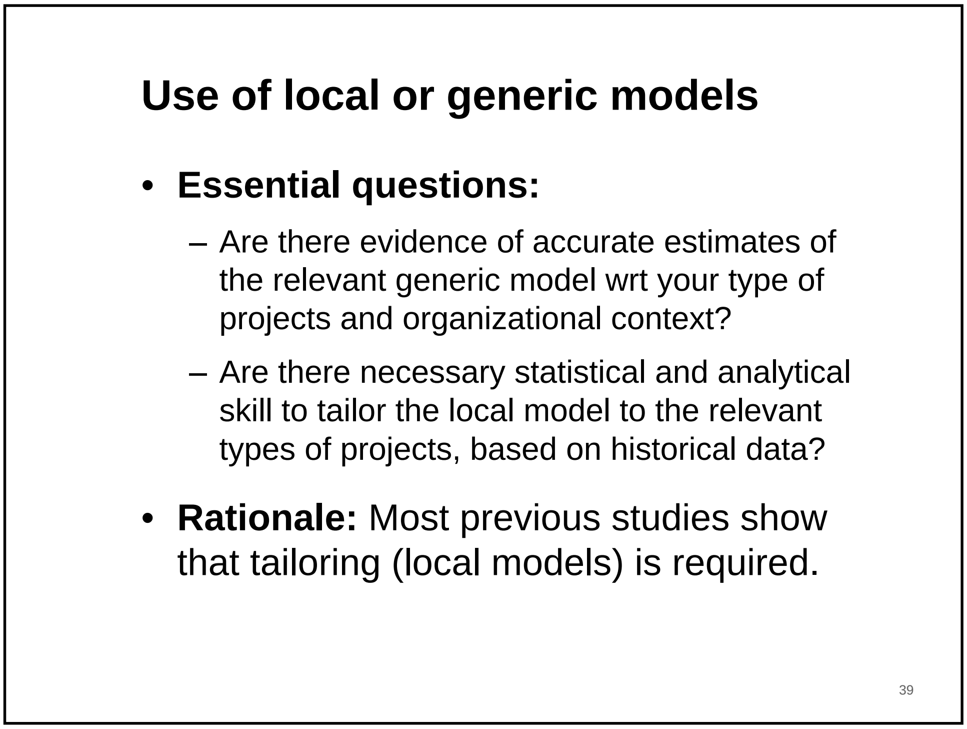
Estimate now or later?

- **Essential question**: Is there sufficient knowledge about the requirement and solution to enable meaningful estimation?
- **Rationale**: Early estimates based on insufficient knowledge may easily become over-optimistic and reduce the organization’s ability to derive realistic estimates when more information gets available due to so-called anchoring.
- Alternatively,
  - estimate only the well-understood parts of the project, or,
  - describe the uncertainty through wide minimum-maximum cost intervals
  - collect more information



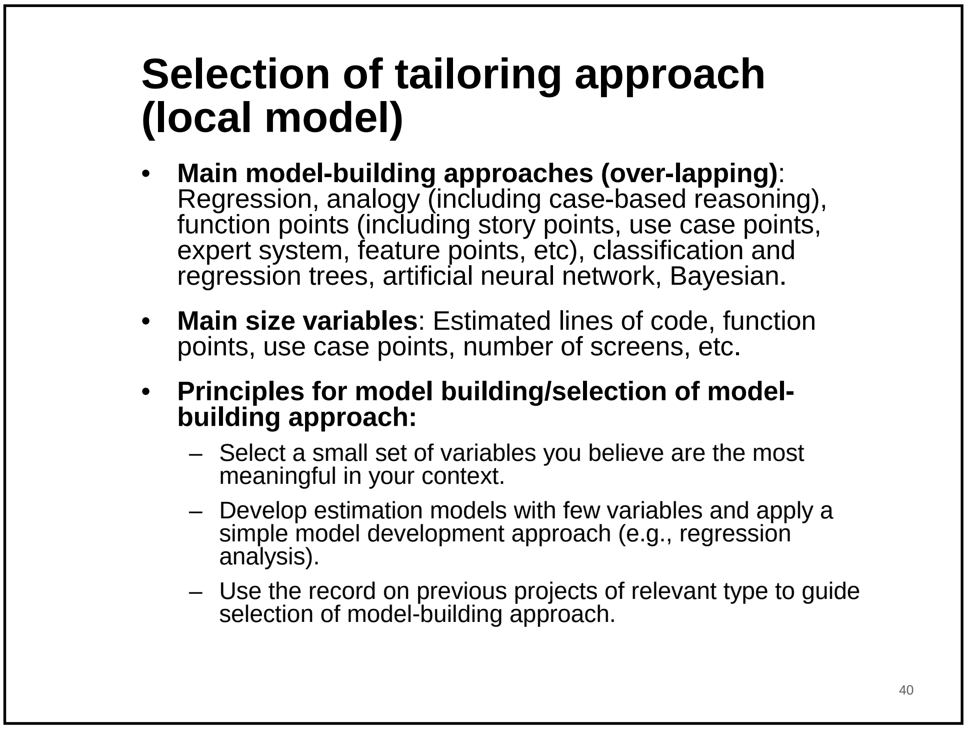
Formal estimation model, expert judgment or both?

- **Essential questions:**
  - Do people in your organization have the necessary statistical and analytical skill to properly use formal estimation models?
  - Is the organization willing to spend effort on implementing, monitoring and, if needed, tailoring the models?
  - Are there important domain knowledge not included in the formal models?
  - Are essential relationships likely to be stable?
  - Is it likely that both formal models and expert estimation provide meaningful estimates?
  - Are there software professionals with experience from similar projects available for estimation?
  - Would you believe in and use the model-based estimate if it diverges substantially from your expert judgment of required effort?



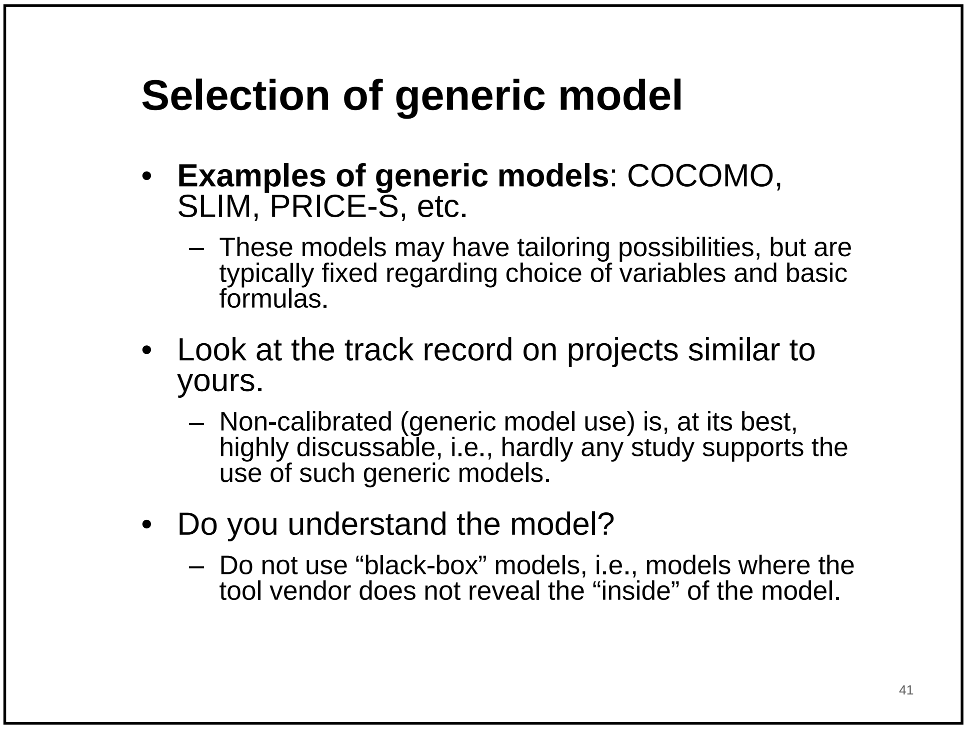
Use of local or generic models

- **Essential questions:**
  - Are there evidence of accurate estimates of the relevant generic model wrt your type of projects and organizational context?
  - Are there necessary statistical and analytical skill to tailor the local model to the relevant types of projects, based on historical data?
- **Rationale:** Most previous studies show that tailoring (local models) is required.



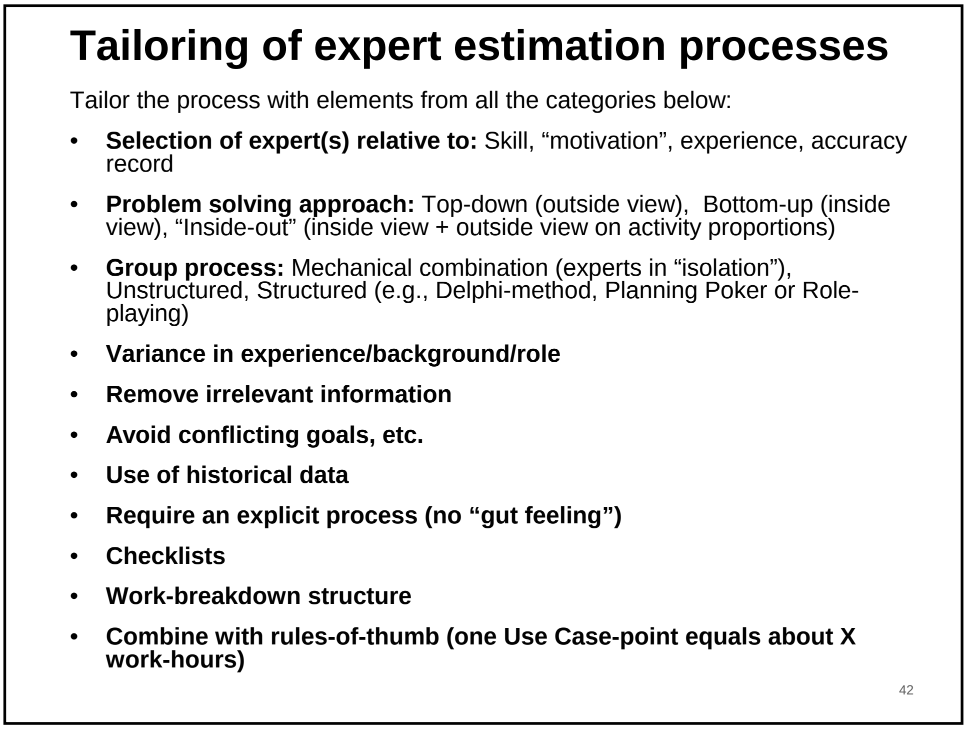
Selection of tailoring approach (local model)

- **Main model-building approaches (over-lapping)**: Regression, analogy (including case-based reasoning), function points (including story points, use case points, expert system, feature points, etc), classification and regression trees, artificial neural network, Bayesian.
- **Main size variables**: Estimated lines of code, function points, use case points, number of screens, etc.
- **Principles for model building/selection of model-building approach:**
  - Select a small set of variables you believe are the most meaningful in your context.
  - Develop estimation models with few variables and apply a simple model development approach (e.g., regression analysis).
  - Use the record on previous projects of relevant type to guide selection of model-building approach.



Selection of generic model

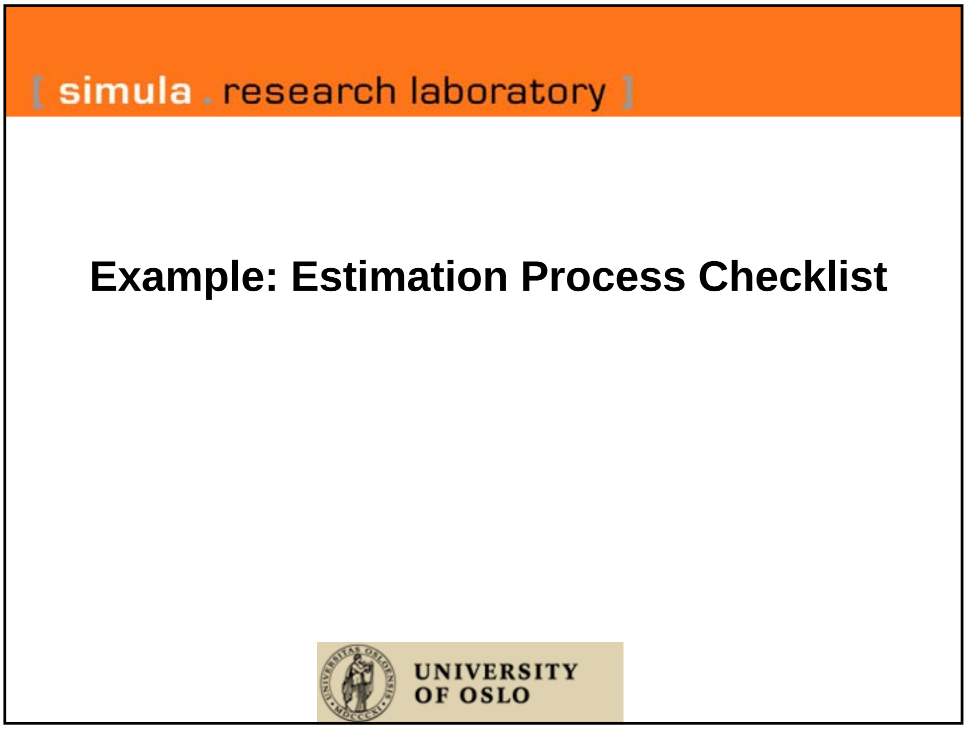
- **Examples of generic models**: COCOMO, SLIM, PRICE-S, etc.
  - These models may have tailoring possibilities, but are typically fixed regarding choice of variables and basic formulas.
- Look at the track record on projects similar to yours.
  - Non-calibrated (generic model use) is, at its best, highly discussable, i.e., hardly any study supports the use of such generic models.
- Do you understand the model?
  - Do not use “black-box” models, i.e., models where the tool vendor does not reveal the “inside” of the model.



Tailoring of expert estimation processes

Tailor the process with elements from all the categories below:

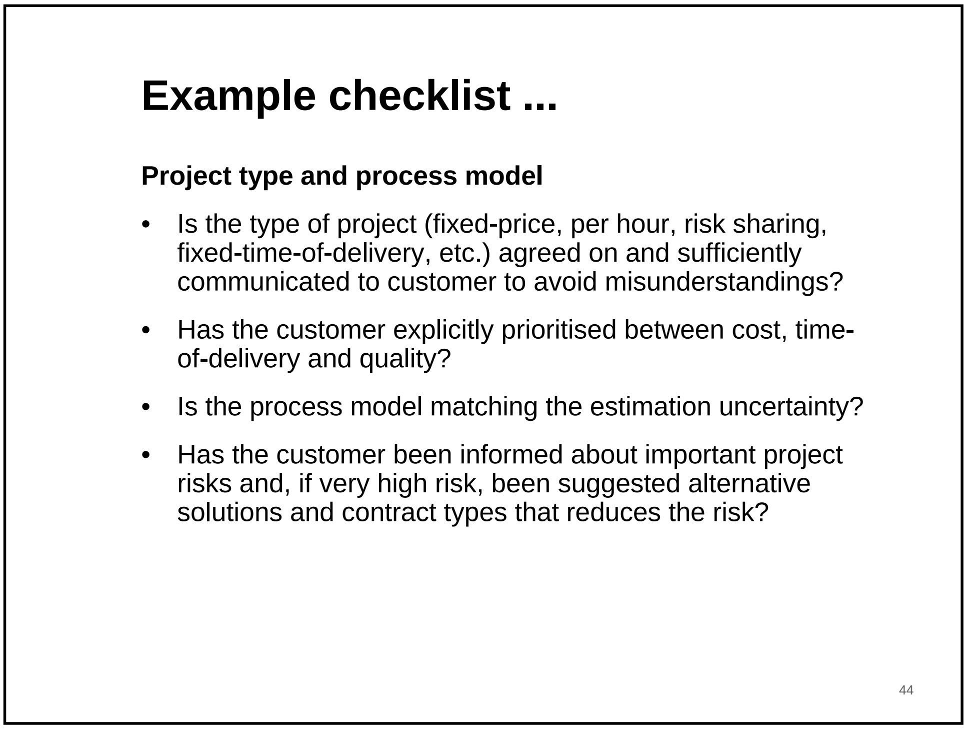
- **Selection of expert(s) relative to:** Skill, “motivation”, experience, accuracy record
- **Problem solving approach:** Top-down (outside view), Bottom-up (inside view), “Inside-out” (inside view + outside view on activity proportions)
- **Group process:** Mechanical combination (experts in “isolation”), Unstructured, Structured (e.g., Delphi-method, Planning Poker or Role-playing)
- **Variance in experience/background/role**
- **Remove irrelevant information**
- **Avoid conflicting goals, etc.**
- **Use of historical data**
- **Require an explicit process (no “gut feeling”)**
- **Checklists**
- **Work-breakdown structure**
- **Combine with rules-of-thumb (one Use Case-point equals about X work-hours)**



[simula . research laboratory]

Example: Estimation Process Checklist

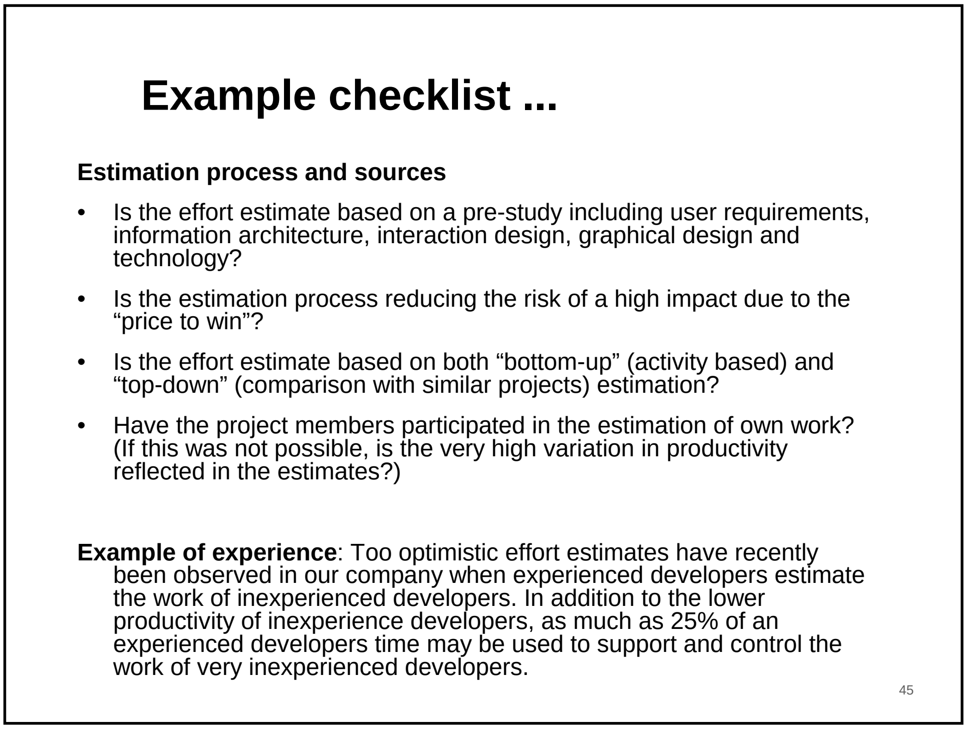
UNIVERSITY OF OSLO



Example checklist ...

Project type and process model

- Is the type of project (fixed-price, per hour, risk sharing, fixed-time-of-delivery, etc.) agreed on and sufficiently communicated to customer to avoid misunderstandings?
- Has the customer explicitly prioritised between cost, time-of-delivery and quality?
- Is the process model matching the estimation uncertainty?
- Has the customer been informed about important project risks and, if very high risk, been suggested alternative solutions and contract types that reduces the risk?

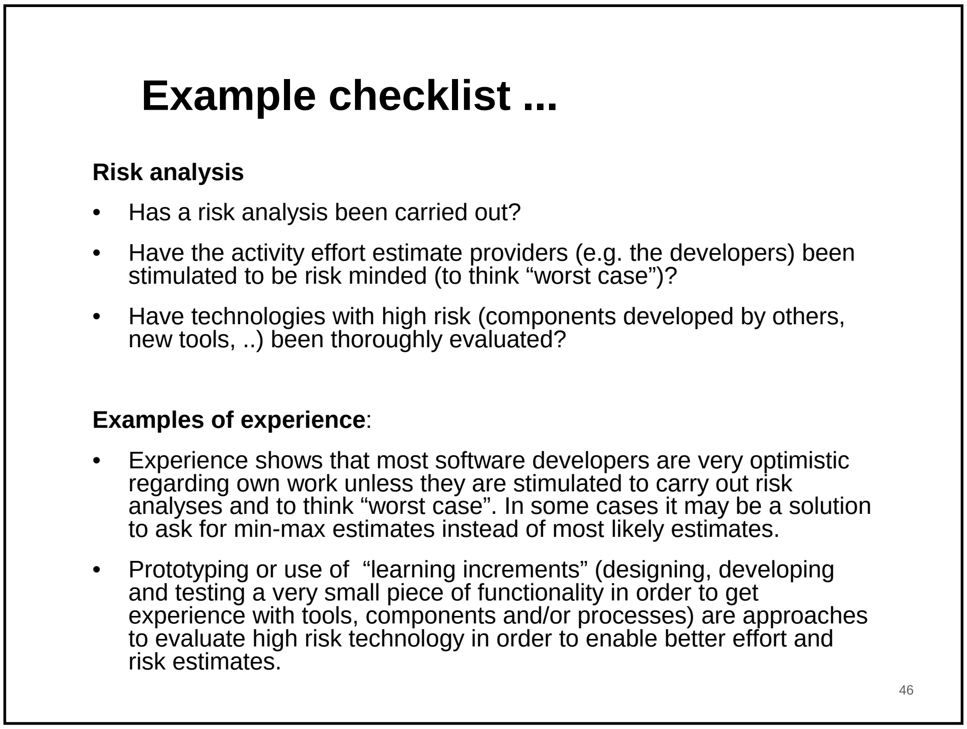


Example checklist ...

Estimation process and sources

- Is the effort estimate based on a pre-study including user requirements, information architecture, interaction design, graphical design and technology?
- Is the estimation process reducing the risk of a high impact due to the “price to win”?
- Is the effort estimate based on both “bottom-up” (activity based) and “top-down” (comparison with similar projects) estimation?
- Have the project members participated in the estimation of own work? (If this was not possible, is the very high variation in productivity reflected in the estimates?)

Example of experience: Too optimistic effort estimates have recently been observed in our company when experienced developers estimate the work of inexperienced developers. In addition to the lower productivity of inexperience developers, as much as 25% of an experienced developers time may be used to support and control the work of very inexperienced developers.



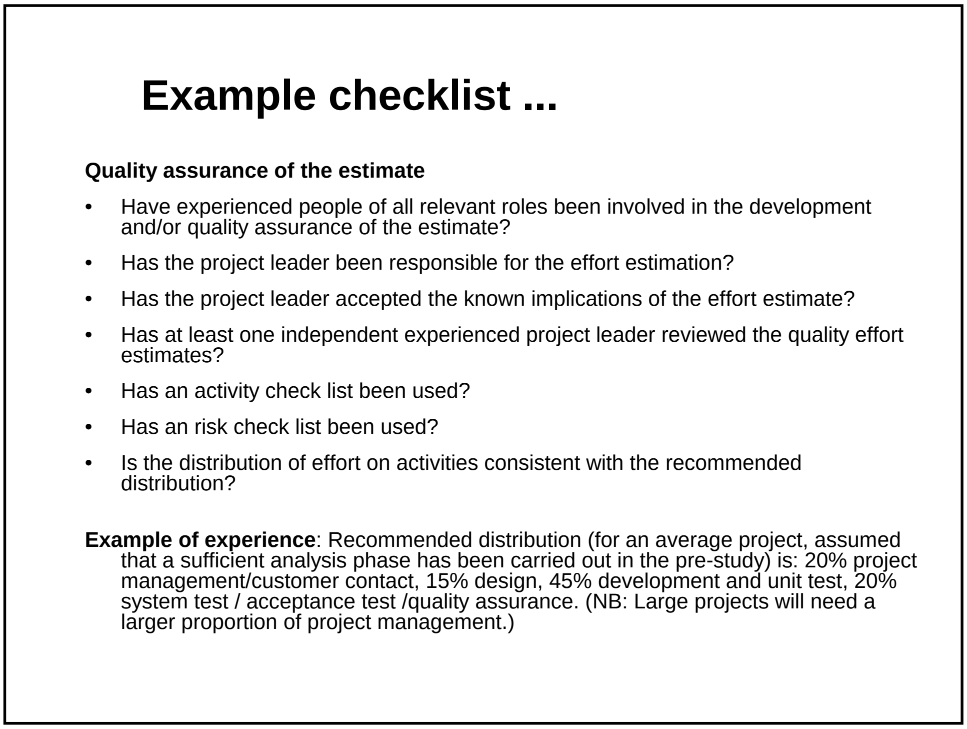
Example checklist ...

Risk analysis

- Has a risk analysis been carried out?
- Have the activity effort estimate providers (e.g. the developers) been stimulated to be risk minded (to think “worst case”)?
- Have technologies with high risk (components developed by others, new tools, ..) been thoroughly evaluated?

Examples of experience:

- Experience shows that most software developers are very optimistic regarding own work unless they are stimulated to carry out risk analyses and to think “worst case”. In some cases it may be a solution to ask for min-max estimates instead of most likely estimates.
- Prototyping or use of “learning increments” (designing, developing and testing a very small piece of functionality in order to get experience with tools, components and/or processes) are approaches to evaluate high risk technology in order to enable better effort and risk estimates.

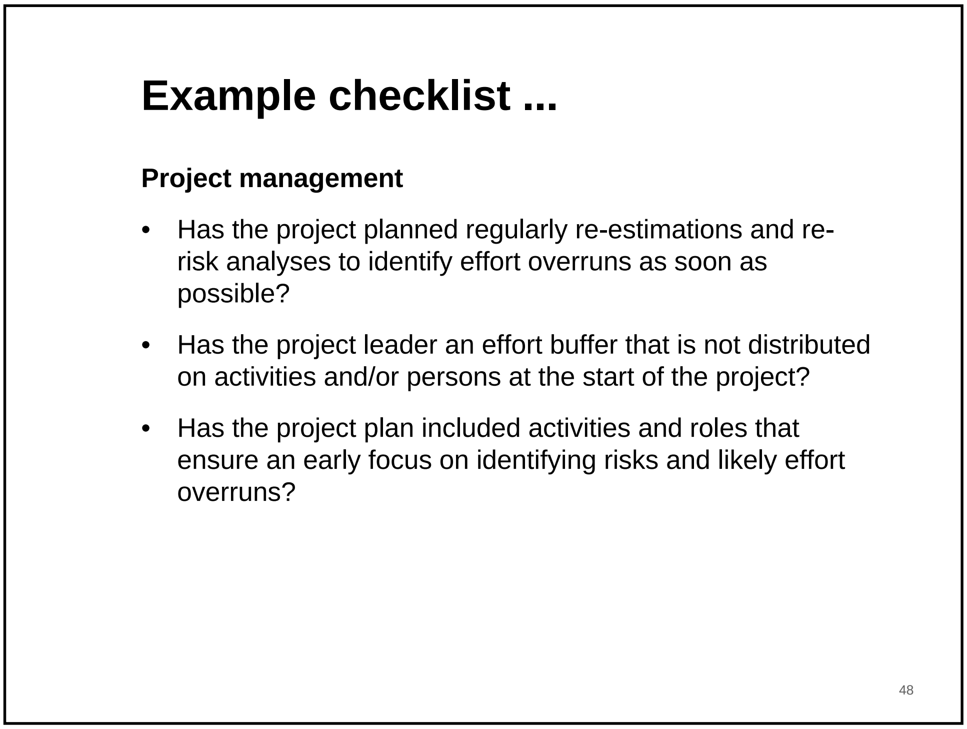


Example checklist ...

Quality assurance of the estimate

- Have experienced people of all relevant roles been involved in the development and/or quality assurance of the estimate?
- Has the project leader been responsible for the effort estimation?
- Has the project leader accepted the known implications of the effort estimate?
- Has at least one independent experienced project leader reviewed the quality effort estimates?
- Has an activity check list been used?
- Has an risk check list been used?
- Is the distribution of effort on activities consistent with the recommended distribution?

Example of experience: Recommended distribution (for an average project, assumed that a sufficient analysis phase has been carried out in the pre-study) is: 20% project management/customer contact, 15% design, 45% development and unit test, 20% system test / acceptance test /quality assurance. (NB: Large projects will need a larger proportion of project management.)



Example checklist ...

Project management

- Has the project planned regularly re-estimations and re-risk analyses to identify effort overruns as soon as possible?
- Has the project leader an effort buffer that is not distributed on activities and/or persons at the start of the project?
- Has the project plan included activities and roles that ensure an early focus on identifying risks and likely effort overruns?

Every company should make their own estimation check list

- Start with the most central issues, i.e., those that have the larges potential for improvement in the software company.
  – Interview 5-10 central project leaders and base the first version on their opinions/experience
- Keep the checklist simple and ”maintainable”
- Make it mandatory to tick off the issues described on the check list
  – ”not relevant” should however be possible, with a brief argumentation

[simula . research laboratory]

Effort estimation uncertainty analysis

UNIVERSITY OF OSLO

Uncertainty: Fortuna, or lottery?

51

Probabilities: A late invention

52

Task: What is the number of inhabitants in Norway

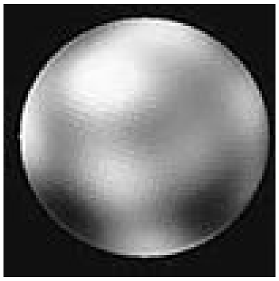
Minimum

Maximum

Be 99% confident to include the correct number in the min-max interval!

What is the radius of the dwarf planet “Pluto”?

Minimum



Maximum

Be 99% confident to include the correct number in the min-max interval!

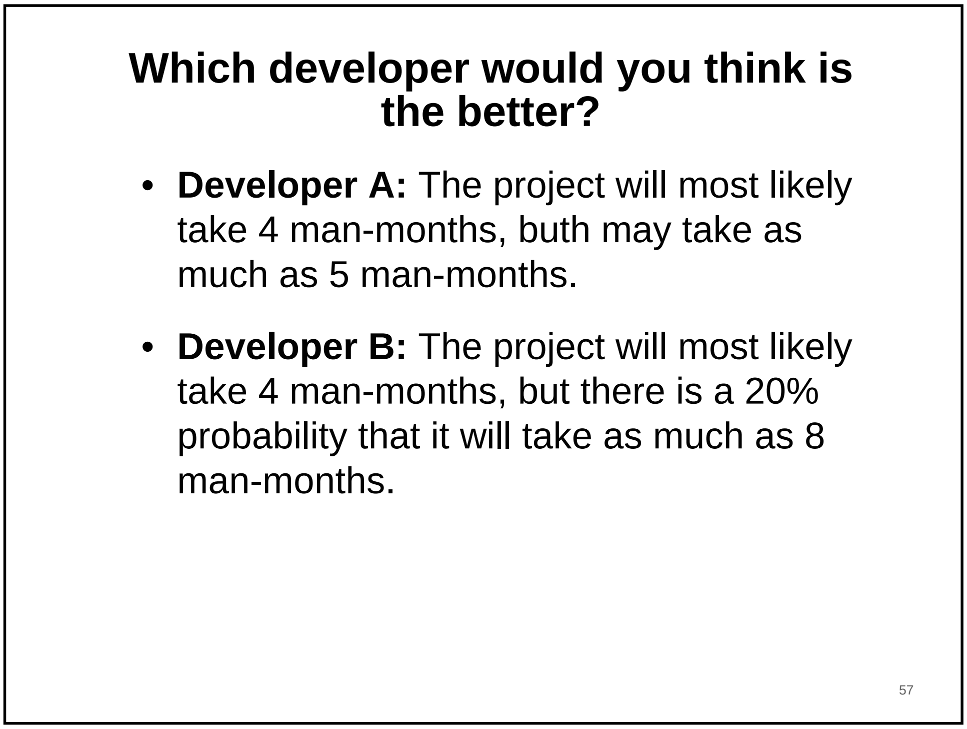
How sure is "almost sure"?

Olympic Games in Montreal in 1976

- *"It can no more lose money than a man can have a baby."* –
- Montreal Mayor Jean Drapeau.

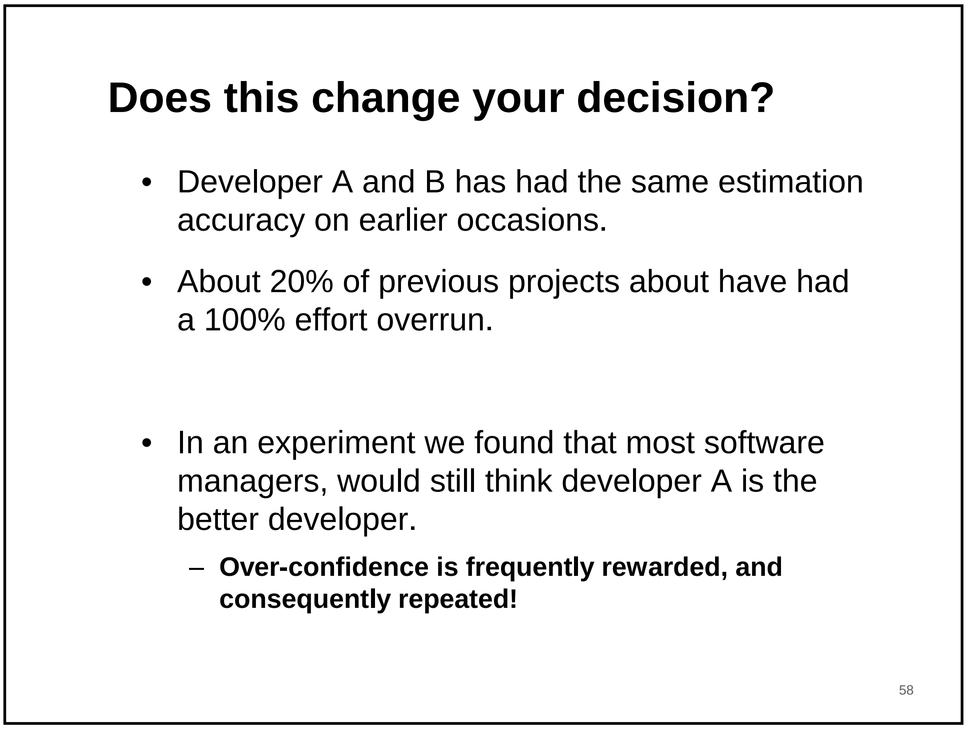
How sure is "almost sure"?

- Our field studies of software companies:
  - Some projects use a minimum-maximum interval method (e.g., PERT)
  - Some did not state how likely they thought it would be to include the actual effort, other assumed a 90% or 98% likelihood.
  - In reality, as much as 40% of the projects was outside the min-max interval!
- In experiments we find that when project managers claim:
  - Almost certainty, this mean about 60% certain
  - "60% certain" = "75% certain" = "90% certain = "99% certain"



Which developer would you think is the better?

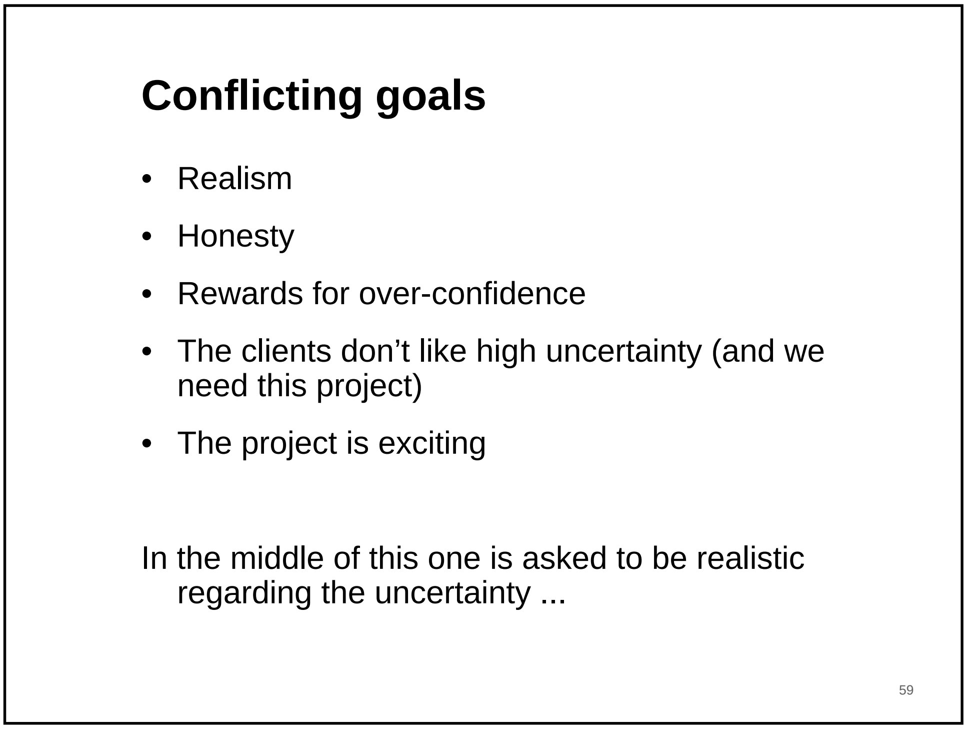
- **Developer A:** The project will most likely take 4 man-months, buth may take as much as 5 man-months.
- **Developer B:** The project will most likely take 4 man-months, but there is a 20% probability that it will take as much as 8 man-months.



Does this change your decision?

- Developer A and B has had the same estimation accuracy on earlier occasions.
- About 20% of previous projects about have had a 100% effort overrun.

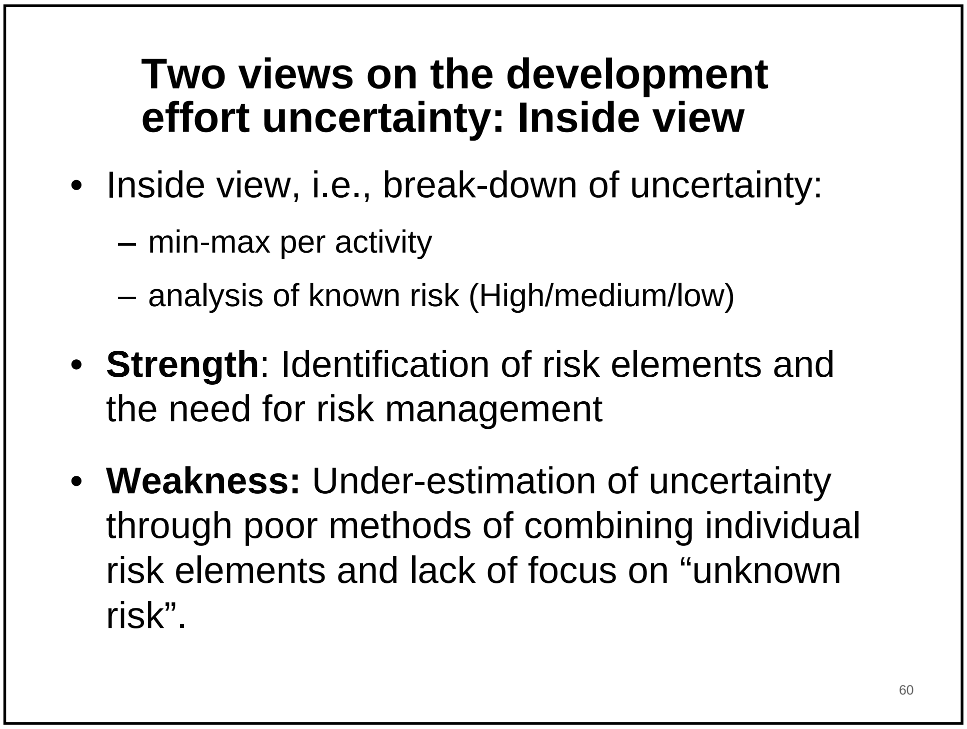
- In an experiment we found that most software managers, would still think developer A is the better developer.
  - **Over-confidence is frequently rewarded, and consequently repeated!**



Conflicting goals

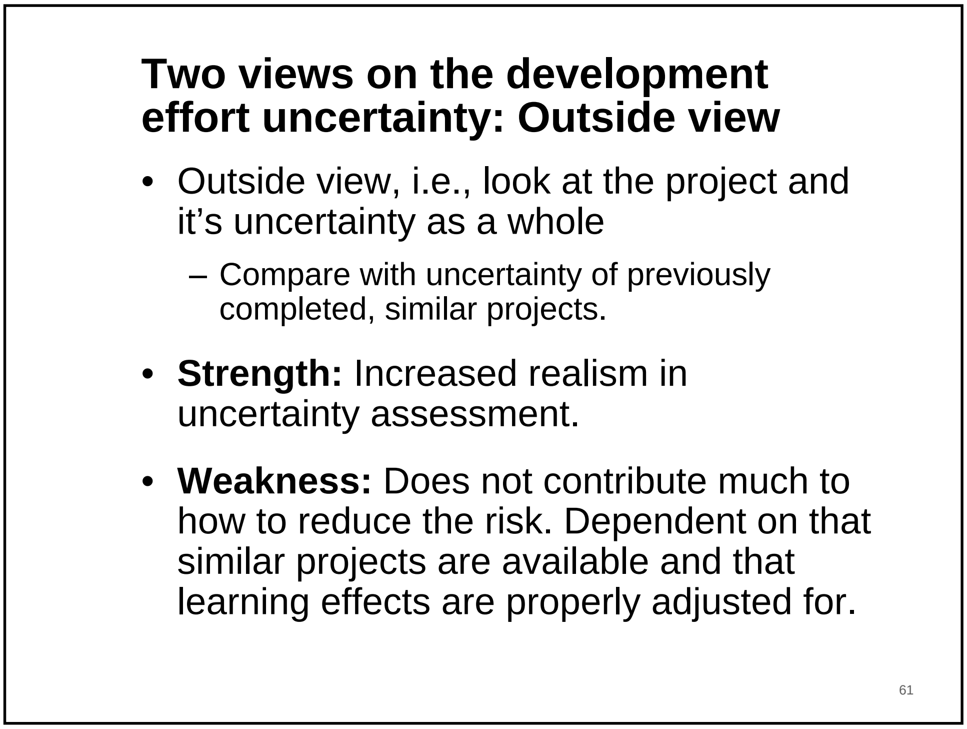
- Realism
- Honesty
- Rewards for over-confidence
- The clients don’t like high uncertainty (and we need this project)
- The project is exciting

In the middle of this one is asked to be realistic regarding the uncertainty ...



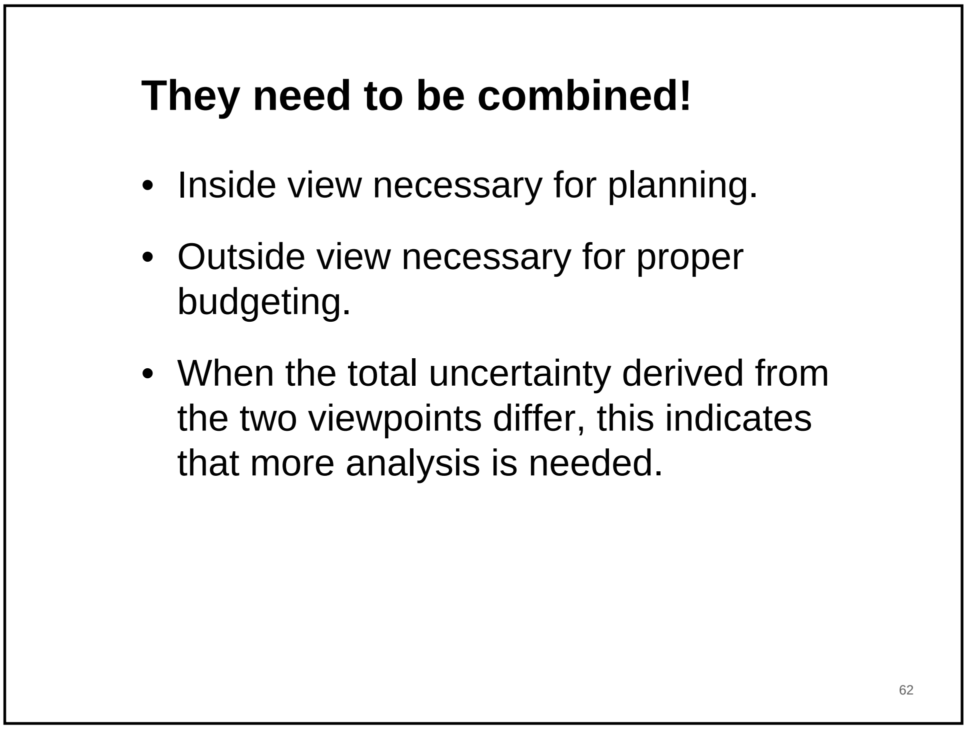
Two views on the development effort uncertainty: Inside view

- Inside view, i.e., break-down of uncertainty:
  - min-max per activity
  - analysis of known risk (High/medium/low)
- **Strength**: Identification of risk elements and the need for risk management
- **Weakness:** Under-estimation of uncertainty through poor methods of combining individual risk elements and lack of focus on “unknown risk”.



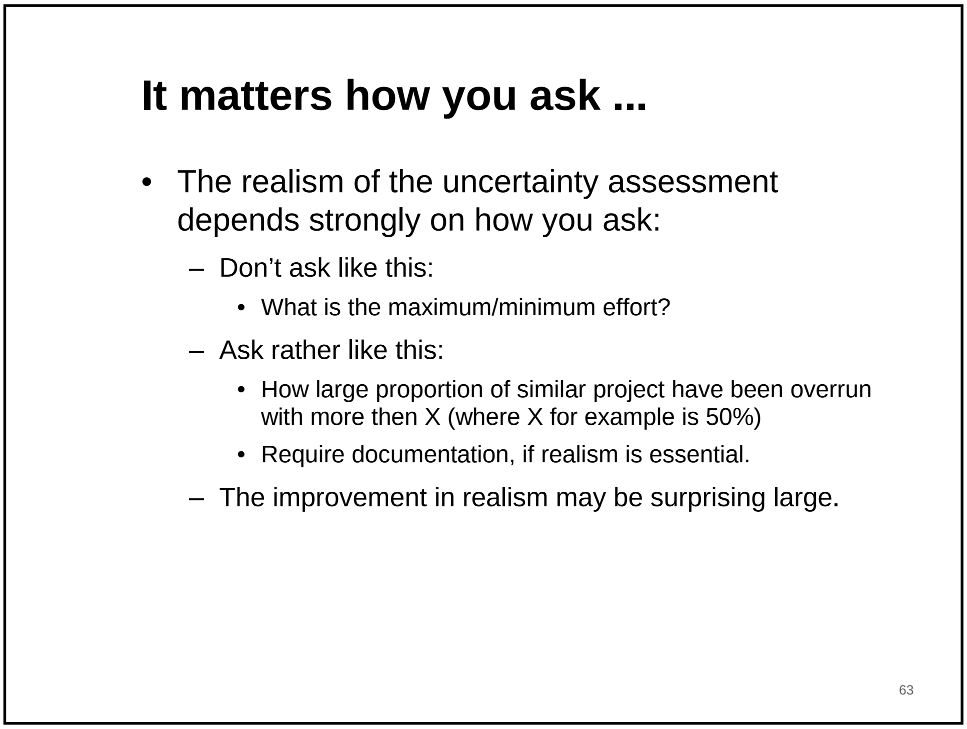
Two views on the development effort uncertainty: Outside view

- Outside view, i.e., look at the project and it’s uncertainty as a whole
  - Compare with uncertainty of previously completed, similar projects.
- **Strength:** Increased realism in uncertainty assessment.
- **Weakness:** Does not contribute much to how to reduce the risk. Dependent on that similar projects are available and that learning effects are properly adjusted for.



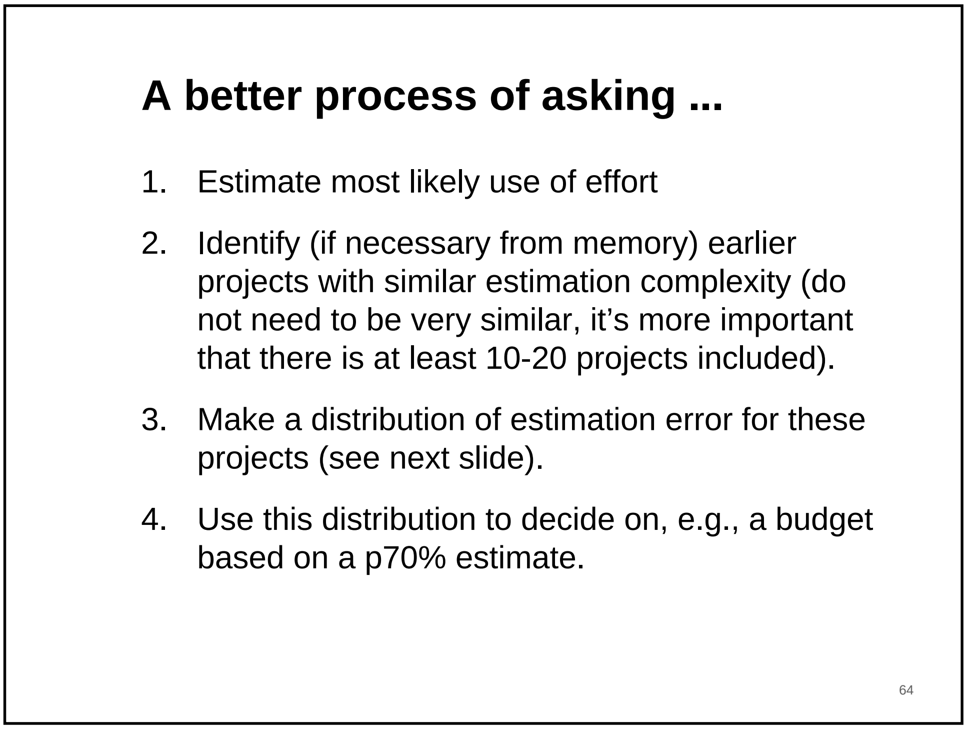
They need to be combined!

- Inside view necessary for planning.
- Outside view necessary for proper budgeting.
- When the total uncertainty derived from the two viewpoints differ, this indicates that more analysis is needed.



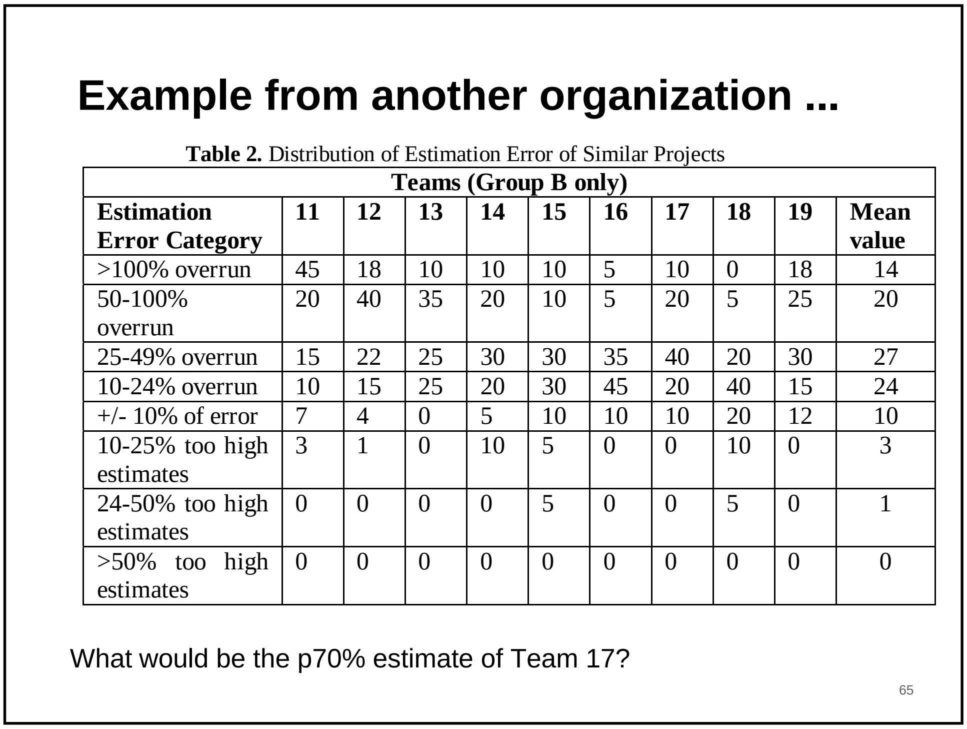
It matters how you ask ...

- The realism of the uncertainty assessment depends strongly on how you ask:
  - Don’t ask like this:
    - What is the maximum/minimum effort?
  - Ask rather like this:
    - How large proportion of similar project have been overrun with more then X (where X for example is 50%)
    - Require documentation, if realism is essential.
  - The improvement in realism may be surprising large.



A better process of asking ...

1. Estimate most likely use of effort
2. Identify (if necessary from memory) earlier projects with similar estimation complexity (do not need to be very similar, it’s more important that there is at least 10-20 projects included).
3. Make a distribution of estimation error for these projects (see next slide).
4. Use this distribution to decide on, e.g., a budget based on a p70% estimate.

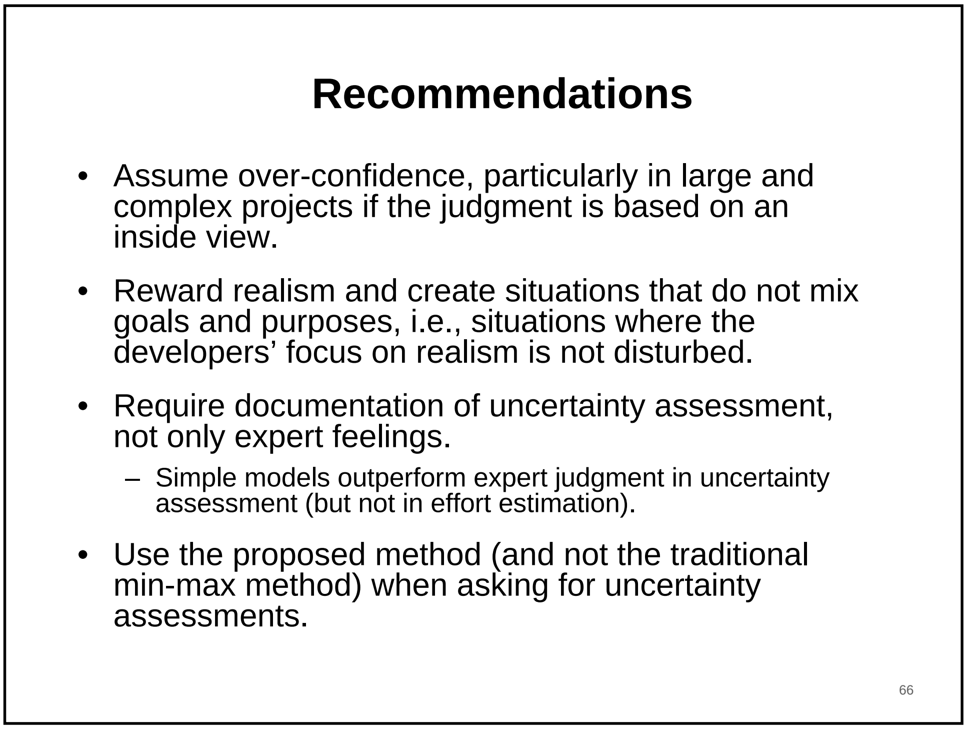


Example from another organization ...

Table 2. Distribution of Estimation Error of Similar Projects

	Teams (Group B only)									
Estimation Error Category	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**Mean value**
>100% overrun	45	18	10	10	10	5	10	0	18	14
50-100% overrun	20	40	35	20	10	5	20	5	25	20
25-49% overrun	15	22	25	30	30	35	40	20	30	27
10-24% overrun	10	15	25	20	30	45	20	40	15	24
+/- 10% of error	7	4	0	5	10	10	10	20	12	10
10-25% too high estimates	3	1	0	10	5	0	0	10	0	3
24-50% too high estimates	0	0	0	0	5	0	0	5	0	1
>50% too high estimates	0	0	0	0	0	0	0	0	0	0

What would be the p70% estimate of Team 17?



Recommendations

- Assume over-confidence, particularly in large and complex projects if the judgment is based on an inside view.
- Reward realism and create situations that do not mix goals and purposes, i.e., situations where the developers’ focus on realism is not disturbed.
- Require documentation of uncertainty assessment, not only expert feelings.
  - Simple models outperform expert judgment in uncertainty assessment (but not in effort estimation).
- Use the proposed method (and not the traditional min-max method) when asking for uncertainty assessments.