



Seminar:  
Software Development Effort Estimation  
Making Waves, Oslo, 8 November, 2007

Magne Jørgensen  
Simula Research Laboratory

**Purpose of seminar**

- Be able to use a precise estimation terminology
- Know important factors that impact the estimation accuracy and/or lead to over-optimism
- Know how to use checklists to improve the effort estimation
- Improve the ability to assess the uncertainty of effort estimates
- Know how to use the agile, group-based estimation technique Planning Poker

## Supporting material

- *"Software effort estimation terminology: The tower of Babel"*
- *"A Framework for Analysis of Software Cost Estimation Accuracy"*
- *"Reasons for Software Effort Estimation Error: Impact of Respondent Role, Information Collection Approach and Data Analysis Method"*
- *"A Checklist for Software Cost Estimation"*
- *"Practical Guidelines for Expert-Judgment-Based Software Effort Estimation"*
- *All material:*
  - Available on-line: [www.simula.no/best](http://www.simula.no/best)

3

## BASIC ESTIMATION KNOWLEDGE



## Estimation error

- Average estimation overrun in IT-projects seems to be about 30%
  - The overrun seems to be the same as in similar engineering disciplines
  - No substantial changes in average estimation error from 1970 until today.
  - The Standish Group's Chaos Report claims that average overrun was 189% in 1994, reduced to 45% in 2000, but is not believable:
    - [www.simula.no/departments/engineering/publications/Jorgensen.2006.4](http://www.simula.no/departments/engineering/publications/Jorgensen.2006.4)
- BUT, what is the meaning of "estimate" and "overrun"?
  - [www.simula.no/departments/engineering/publications/Grimstad.2006.1](http://www.simula.no/departments/engineering/publications/Grimstad.2006.1)

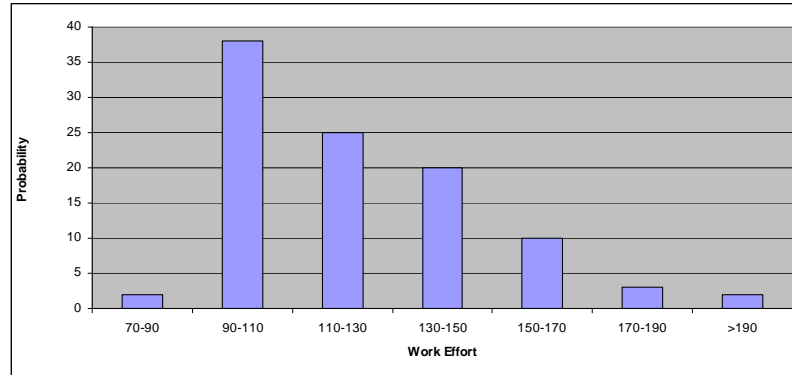
5

## What is an "estimate"?



6

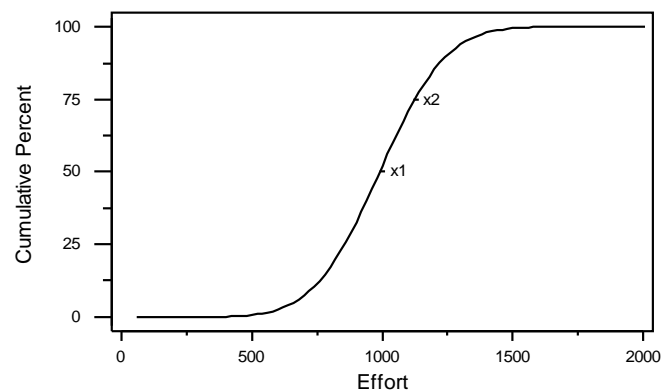
## Effort Usage Probability Distribution



ESTIMATE = Most likely effort, 50% estimate, most optimistic effort, ideal effort, 70% estimate, planned effort, budgeted effort, priced effort, or, ...?

7

## S-curve: Work Effort



8

## Estimation Error (Overrun)

- Measure of difference between actual and estimated effort
  - Be precise and consistent with respect to what you mean and try to communicate by an effort estimate
- Estimates are seldom "correct"
  - A 50% estimate will 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

9

## Recommendation

- Use different terms for different concepts/purposes:
  - Most likely use of effort. Purpose: Realism, and just that.
  - Planned use of effort (e.g., 70%-estimate). Purpose: Control of project. May include contingency buffers.
  - Budget (e.g., based on 80%-estimate). Purpose: Financial control of project portfolio.
  - Price (e.g., based on 40%-estimate). Purpose: Long or short term win/loss considerations.
- Different purposes should lead to different processes. Realism and market considerations (e.g., winning a bidding round) at the same time, means that realism will suffer.

10

## Reasons for Estimation Error

### Estimation accuracy is impacted by (1)

#### 1) *Estimation ability*

- *Relevance of earlier experience*
- *Ability to select appropriate estimation strategy*
- *Ability to use estimation model or method correctly*

## **Estimation accuracy is impacted by (2)**

### *2) How difficult it is to estimate*

- *Project management ability*
- *Development skill of team members*
- *Relation to client*
- *The clients' ability to do their part of the project*
- *Completeness and correctness of information (e.g., the requirement specification)*
- *Inherent completion complexity*
- *Project priorities (cost, time-to-market, quality, ...)*
- *Flexibility of product and process*

13

## **Estimation accuracy is impacted by (3)**

### *3) How you measure*

- *What actual effort is compared with*
  - *Estimate = most likely effort, planned effort, budget, 80%-estimate ....*
- *Quality of actual effort data*
  - *Over-time included?*
- *Difference between planned and actual product and/or process*

14

## Results from a study

- “Reasons for Software Effort Estimation Error: Impact of Respondent Role, Information Collection Approach and Data Analysis Method”, *IEEE Transactions on Software Engineering*
- Main message is that it is difficult to analyze reasons for estimation error:
  - Direct reasons are typically emphasized
  - Indirect reasons are forgotten
  - “Systemic” reasons are seldom analyzed
- 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.

15

## Recommendation

- Accept inaccurate 50%-estimates if:
  - The estimation work process is of high quality
  - The estimation complexity is high
  - The estimation uncertainty is well communicated
  - The plans and budgets are based on a realistic view on the risks
- Reward realistic uncertainty assessments
  - The opposite happens most of the time, indirectly
  - Accept that risk cannot always be managed, but must be accepted. Flexibility in product, process and budget should be present.
- Analyse reasons for overruns, but do not forget to analyse reasons to accurate estimates:
  - Be realistic about what you can learn about reasons for overruns
  - Ask “why” five times
  - Apply “Root cause analysis”

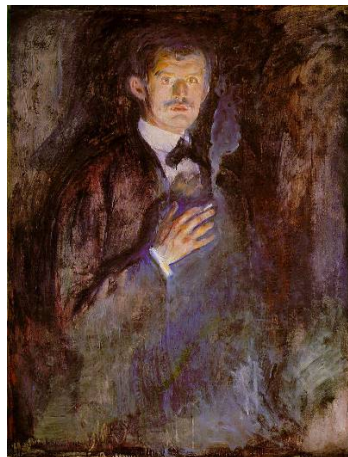
16

**Why are we over-optimistic again and again?**



17

**If you need a realist, look for slightly depressive people.**



18

**Overconfidence (under-estimation of risk) is normal**



19

We need  
optimists –  
but not  
necessarily  
as estimators



20

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



21

## Reasons for this over-optimism

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

22

## Biology



- Evolution rewards over-confidence and over-optimism (Girls: You are to blame. You select the most over-confident boys.)
- 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.

23

## Culture

- There are cultural differences in level of optimism, but the findings are confusing.
  - Asian people (collective oriented) seem to be more over-optimistic than Western people (individual oriented)
  - Studies on "self-efficacy" found that Western people were more over-optimistic than Asian people.
  - A study on the "planning fallacy" found no difference in degree of over-optimism.
- Culture may be less important than many people believe in explaining over-optimism.



24

## 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.



25

## Motivation (cognitive dissonance)

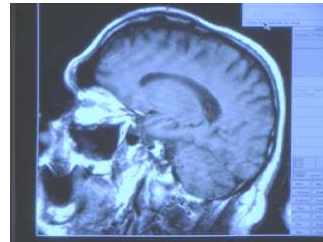
- A good self-evaluation is beneficial
  - For yourself
  - Because it's used as a performance indicator by others. Derfor bør ikke et prosjekt ta lang tid, det bør ikke oppstå problemer man ikke kan løse, ...
- Low effort estimates = high performance = better (but less realistic) self-evaluation.
  - Otherwise, we have a cognitive dissonance, i.e., a difference between what we estimate and who we want to be.



26

## 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?



27

## 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

28

## Bidding round process

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

29

## 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.

30

## Checklists

### Checklists

- Based on: *"A Checklist for Software Cost Estimation"*
- The checklist should be tailored to the organization using it (and not be too long)
- Addition to existing estimation process
- Updated as an experience database

## **Part 1: Estimation Preparation**

- 1. Understand the estimation problem
  - Identify the purpose and the accuracy requirements
  - Identify stake holders and political issues
- 2. Ensure agreement on purpose and essential assumptions
  - Identify relevant decisions and assumptions that can have a significant impact on the estimation work
  - Decide whether it is meaningful to estimate or not
  - Identify level of flexibility in process and product and project priorities

33

## **Part 1: Estimation Preparation**

- 3. Collect relevant information
  - Remove irrelevant information
  - Identify the main cost drivers
  - Ensure that the information sources are neutral
- 4. Select estimation process
  - Base the estimation process on success on similar, previously completed projects

34

## **Part 2: Estimation Phase**

- 5. Start with the estimation of most likely effort
  - Structure the estimation process with checklists, pre-defined templates for work-break-down structures, etc.
  - Describe all essential assumptions you make
  - Describe your use of historical data
- 6. Assess the uncertainty of the estimate of most likely effort
  - More on this later.

35

## **Part 2: Estimation Phase**

- 7. Review the estimation process and the effort estimate
  - Use independent experts for this
  - Develop a review checklist

36

### **Part 3: Application of the estimates of most likely effort**

- 8. Bidding
  - Estimates of most likely effort and data from the uncertainty assessment will show the likelihood of profit for a given bid. Use this to decide whether a participation in the bidding round is worthwhile.
- 9. Planning
  - Decide, based on historical data, on the need for a buffer for unexpected events.
  - Plan re-estimation

37

### **Part 3: Application of the estimates of most likely effort**

- 10. Communicate estimates, bids, plans and uncertainty
  - Adapt the information (e.g., about uncertainty) to the maturity of the receiver
- 11. Control the costs
  - Monitor the development and re-estimate
  - Keep the monitoring processes as simple as possible

38

## **Part 4: Learning from experience**

- 12. Learning
  - Try to understand the deeper (indirect) reasons for estimation problems
  - Update the check list (the experience data base), the estimation process and the templates on basis of experience (a job for a cost engineer?)
  - Try to avoid learning fallacies

39

## **Make your own check list**

- Start with the most central issues, i.e., those that have the largest potential for improvement in your organizations.
  - 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 be possible, with a brief argumentation

40

## Effort estimation uncertainty analysis

### Estimation uncertainty

- We find that when project managers claim:
  - Almost certainty, this mean about 60% certain
  - “60% certain” = “75% certain” = “90% certain = “99% certain”
- 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.

## The process

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.

43

## Example from another organization ...

**Table 2.** Distribution of Estimation Error of Similar Projects

Estimation Error Category	Teams (Group B only)									Mean value
	11	12	13	14	15	16	17	18	19	
>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?

44

## Group estimation

### Experiment: individual vs group estimation [\*]




- Twenty software professionals with different backgrounds provided individual effort estimates of a software development project.
  - Real-world project that had been completed
- Then they formed five estimation groups. Each group agreed on a project effort estimate
  - Through discussion and combination of knowledge.

[\*] Moløkken-Østfold and Jørgensen (2003): Software Effort Estimation: Unstructured Group Discussion as a Method to Reduce Individual Biases. In The 15th Annual Workshop of the Psychology of Programming Interest Group

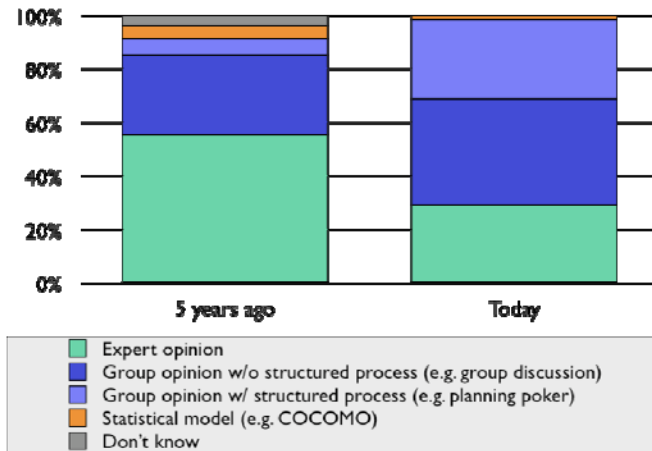
- The group discussion-based estimates were closer to the effort used by the actual project than the average of the individual experts
  - Possible explanation: the groups' ability to identify more project activities
  - Possible explanation: justification increase realism
- Similar results obtained in experiments on uncertainty intervals [\*]
  - Group discussion lead to more realistic uncertainty intervals

## Research on group estimation

- Few studies in a software cost estimation context
    - ...but many relevant studies in other research fields (psychology, business forecasting, etc)
  - Findings
    - Combination of estimates improve the estimation (especially when the estimators have different background)
    - Structure can improve the estimation (e.g. remove the impact of irrelevant information)
    - "More heads remember more"
  - Disadvantages
    - Resource demanding (expensive) compared to individual estimation
    - "Group think" can occur (e.g. everybody agrees with the leader)
    - "Group polarization" can occur (e.g. the group is more optimistic than the average of the individuals)
- 
- A cartoon illustrating the concept of groupthink. On the left, a man in a suit stands and addresses a group. Three men are seated at a table, looking at him. Each has a thought bubble above their head. The first bubble says "SAY IT ANY! GO!". The second bubble says "YOU'VE GOT TO BE RIGHT!". The third bubble says "FEELING THE SPIRIT!". This visualizes the pressure to conform and the loss of individual critical thinking in a group setting.



### Group estimation is gaining popularity in the Norwegian IT industry (survey at JavaZone 2007)



49

### Structured group estimation's impact on perceived estimation accuracy (JavaZone 2007)

- 50% believed that their estimation accuracy was improved
- 30% believed that their estimation accuracy was unchanged
- 10% believed that their estimation accuracy was worsened
- 10% did not know

50

## Structured group-based estimation methods

- Planning Poker
- Wide-band Delphi



51

## Planning Poker

- Agile estimation technique
  - Described by Grenning [1] and Cohn [2]
1. Customer explains story
  2. Team discusses work involved
  3. Each estimator picks a card representing estimate
  4. Everybody reveals estimate simultaneously
  5. Lowest and highest estimator justifies
  6. Team discusses the estimates
  7. Repeat from 3. until estimates converge
  8. Team decides on collective estimate

[1] J. W. Grenning, *Planning Poker*, 2002  
[2] M. Cohn, *Agile Estimating and Planning*, 2005

52

## When to use Planning Poker?

- Release planning
  - customer picks features for next release
  - estimates basis for prioritising features and staffing
  - planning poker quickly provides realistic estimates and reveals unclear requirements
- Detailed planning (iteration) and design
  - breaking features into tasks and assigning responsibility
  - estimating with planning poker reveals unclear requirements
  - planning poker can facilitate design discussion

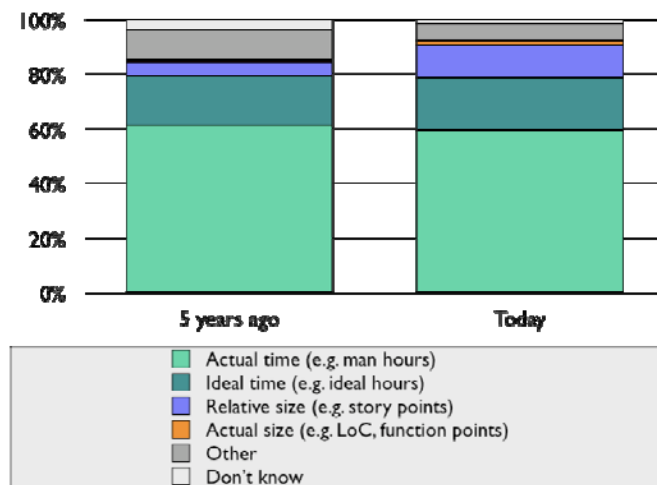
53

## What should we estimate?

- Estimate size, not duration
  - Easier to discuss
  - Aim for consistency
- Alternative units for size
  - Story points
  - Ideal days
- Derive duration when planning
  - Measure project velocity and apply “yesterdays weather”
  - Project velocity = sum of estimation points for user stories completed in iteration

54

## What do you typically estimate?



55

## Should we use fixed or flexible sizes?

- Fixed sizes easier and more effective
  - Experiments with flexible sizes indicates that the group tends to standardise anyway
  - Less options speeds up the process
  - Fibonacci-sequence is effective: 1, 2, 3, 5, 8, split
- Remember: these are estimates
  - We don't need the added precision flexible estimates might give us
  - A few hours give or take is usually of minor importance

56

## **Should we seek consensus or go with the average?**

- Justify estimates after first round of planning poker
  - Reveals what people take into account in their estimate
  - Important for revealing further details
- Recommendation
  - Always do at least two rounds of planning poker
  - Continue doing rounds as long as individual differences in estimates are big
  - Average or go with majority when differences in estimates are small

57

## **Things to watch out for when doing planning poker**

- Over-timing it and getting into too much detail
  - Don't discuss too long before doing first round of planning poker
  - After a certain point further discussion adds less value
  - Use timer if long discussions become a problem
  - Remember these are estimates
- Not getting different viewpoints
  - A lot of questions will pop up during discussions
  - Beneficial to have multiple points of view present when making assumptions

58

## Why does planning poker (supposedly) work?

- Simultaneous display of estimates reduces bias
  - The first estimate suggested will usually create an anchor
  - Some people have more influence
- More questions asked and more information shared
  - More people bring more knowledge to the table
  - Different people consider different aspects
- Broader range of developers provide estimates
  - Research has shown that combining estimates reduces over-optimism
  - Different people employ different estimation strategies
- Estimates better reflect team's average ability to solve task
  - Expert estimates tend to be based on expert abilities
  - You do not know who on the team will actually end up doing the task

59

### Industrial studies

	Planning poker vs. unstructured group	Planning poker vs. individual expert
Planning scale	Release planning (2-3 months)	Sprint planning (2 weeks)
Team	8-12 developers	4-6 developers
Automated acceptance tests	Yes	No
Pair programming	Yes	No
Progress visibility	Story cards on wall	Jira
Customer view in session	Business analyst	Developers

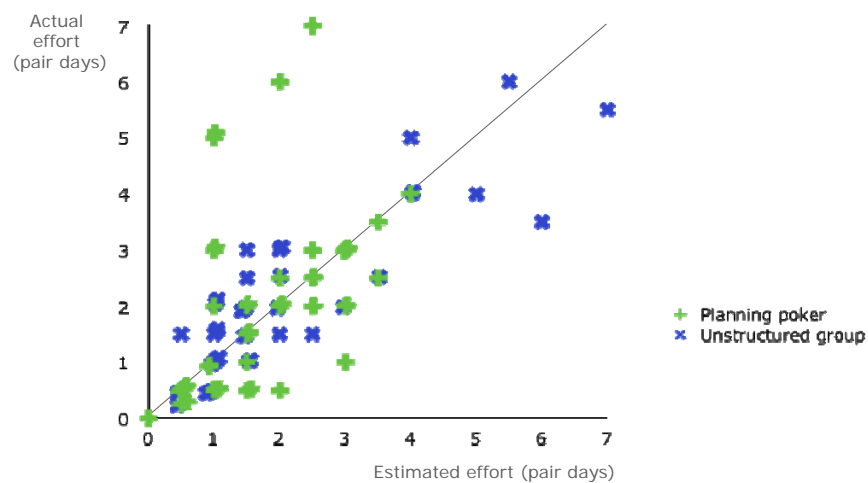
60

## Common for both studies

- Fun! Both teams carried on using it
- More efficient estimation process
- Increased ownership of estimates
- Increased responsibility for the project progress
- How did it affect estimation accuracy?

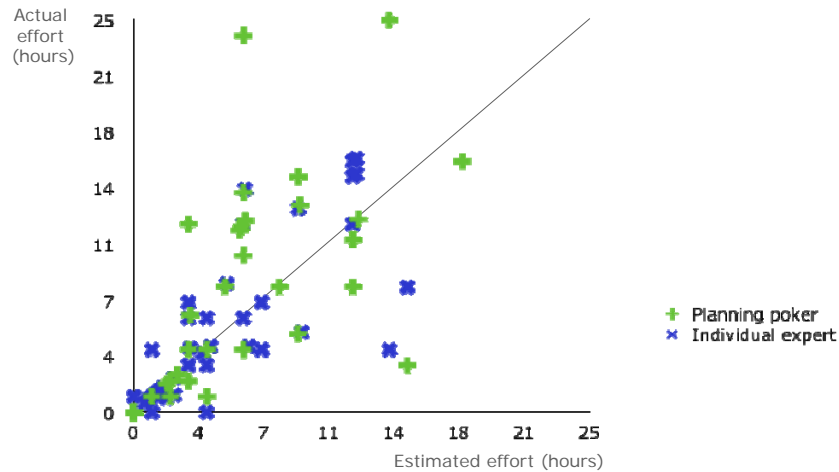
61

## Planning poker vs. Unstructured group estimation



62

## Planning poker vs. Individual expert estimation



63

## Wideband Delphi (example)

1. Preparation of estimation process
  - Create estimation material
  - Select estimation personell including moderator
2. Kick-off meeting
  - Moderator presents the estimation problem, the estimation material, the estimation process, the estimation units, etc
  - The group discuss selection of experts, estimation material, etc
3. Individual estimation
  - Identify activities and estimate
  - External experts can be consulted
4. Estimation meeting
  - Moderator present summary of all estimates and activities
  - Experts discuss the results (focus on anonymity)
5. Summary
  - Often done by moderator and project leader

64

## Final words

### Implications for you?

- What should you change?
  - More precise use of estimation terminology?
  - Estimation process support? (Cost engineer?)
  - Improvement of check lists?
  - More use of group based effort estimation (e.g., Planning Poker)?
  - Better and more systematic learning from experience?
  - Uncertainty assessment based on historical data?
  - Other issues?