[simula . research laboratory]

Software Development Effort Estimation: Why it fails and how to improve it

Ho Chi Minh city, December 2010

Magne Jørgensen
Simula Research Laboratory & University of Oslo

UNIVERSITY OF OSLO

About me

- Scientific researcher at Simula Research Laboratory, Oslo, Norway
  - prof. at Univ. of Oslo
  - Research reports can (free of charge) be downloaded from: *simula.no/research/engineering/projects/best*
  - Extensive industrial experience as programmer, project manager, process improvement managers and general manager.
  - Responsible for estimation work and training in several companies.
- Conduct advisory work and seminars for software companies.

BASIC EFFORT ESTIMATION KNOWLEDGE

UNIVERSITY
OF OSLO

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*

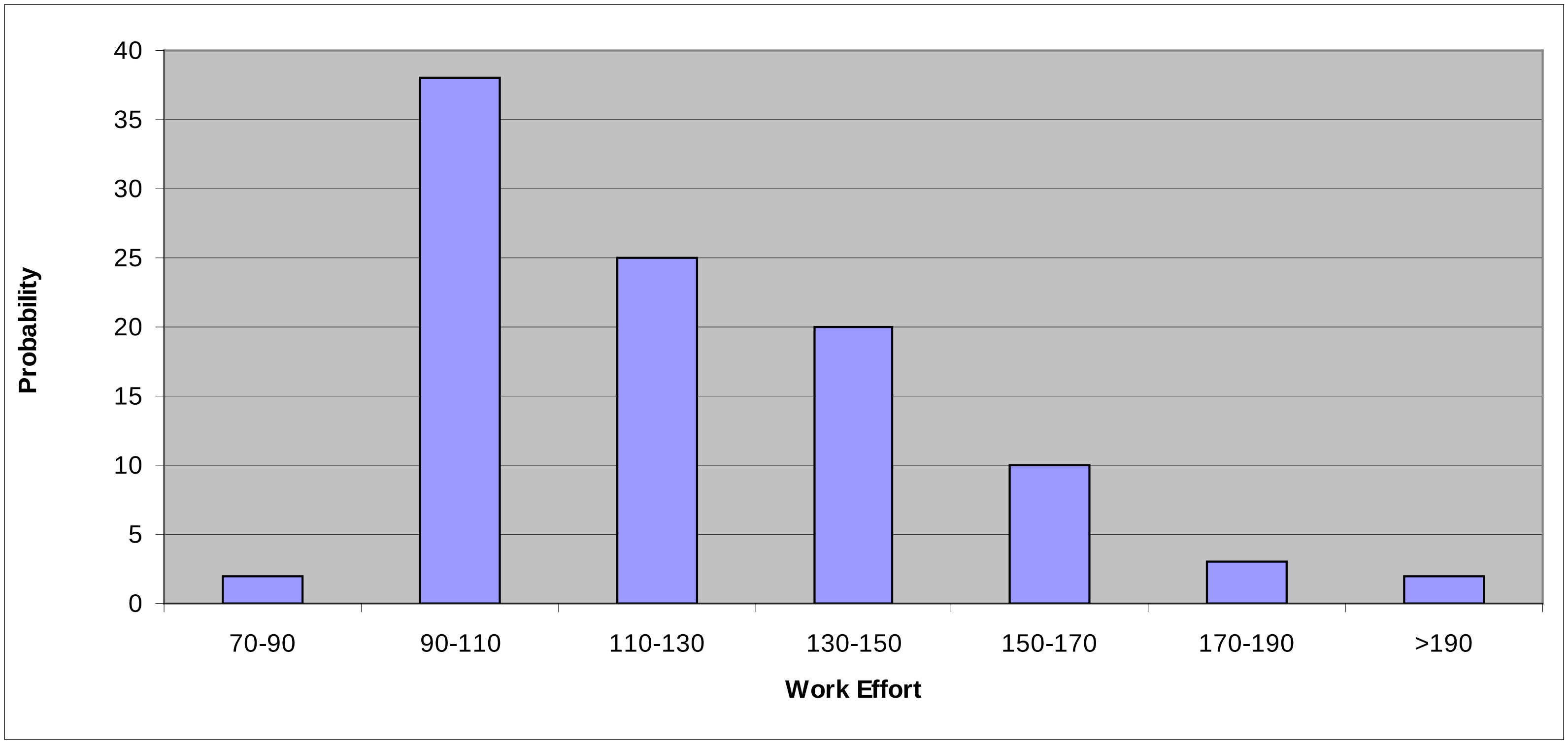
Estimation error

- 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”?

Do we know what we mean by ”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.

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!

Reasons for Estimation Error
(and how to improve the processes)

UNIVERSITY OF OSLO

The better-than-average effect….

Over-confidence …

Motivation

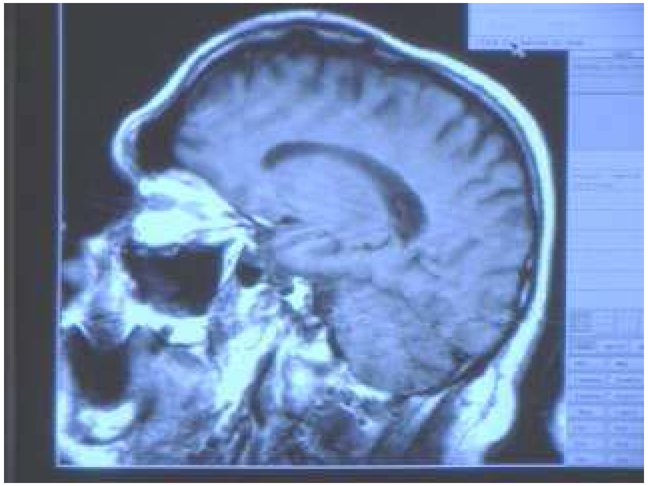
- Mix of “I hope this does not take more than …” and “This will not take more than …”
- Optimism can have a positive impact on performance, BUT
  - Only for a short period of time.
  - The effect is over-rated.

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?



Bidding round format frequently leads to over-optimism

- The winner's curse
  – You only win bidding round when being over-optimistic.
- Bidding anchors
  – Budget
  – Early price indications
  – Expectations

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]

When Should We Trust Expert Judgment in Software Development?

UNIVERSITY OF OSLO

Who are the Experts?

- Those with long experience?
- Those with accurate judgments?
- Those with high confidence in their judgment?
- Those with the best skill, knowledge and/or process?
- This with highest CWS-index? (CWS Cochran-Weiss-Shanteau)
  - *CWS -index = discrimination / consistency*
- Those recognized as experts by at least one other person? (or people away from home, such as me ….?)
- U.S. Supreme Court classifies legal experts in Federal Rule of Evidence 702 as:
  - *”individuals with scientific, technical, skill, experience, training, or education that will assist the trier of fact* [judgment of facts] *to understand the evidence or to determine a fact at issue.”*

Some Expert Characteristics ...

- Experts excel mainly in their own domain (expertise is narrow)
- Experts has a large knowledge base, e.g., consisting of chunks (more than 10,000?), rules and schemata.
- The experts perceive large meaningful patterns in their domain (e.g. identify chunks stored in their knowledge base)
- Experts see and represent a problem in their own domain at a deeper (more principled) level than novices; novices tend to represent a problem at a superficial level.
- It takes at least 10 years with “deliberate practice” to achieve top performance.
- Experts do not differ from non-expert in basic information-processing power, but mainly in amount of “deliberate practice”.

For an overview, see, for example: *Expertise, models of learning and computer-based tutoring*, by F. Gobet and D. Wood, 1999.

Example: Judgment-based effort estimation.

- Ask a software professional about his judgment-based estimation process or use a think-aloud protocol to collect this information, and you will NOT get much valuable information.
  - They typically respond with “don’t know”, “it felt right” or present vague statements about their use of experience.
  - The may also feel that they should know how they did the estimation work, and start to rationalize, e.g., by describing how they believe they should have done this as rational beings.
- The same goes, I guess, for expert-judgment based assessment of properties like “maintainability”, “user friendliness” and “quality”.
- It is consequently not possible to gain much insight into these expert judgment-based processes by asking people (think-aloud protocols, interviews, experience reports) or observing their actions. (We have tried and failed several times ...)

The dual theory of cognition (analysis vs. intuition)

- Suppose that we have a simple model, e.g., the rule that a medium complex “use case” takes 8 work-hours.
- Use of that model implies that a task with five medium complex use cases should take about 40 work-hours.
- The estimator, however, feels that 40 work-hours is too high, and, that 30 work-hours should be sufficient. We now have a conflict between analysis and intuition.
- We tend to have more confidence in the analytical **process**, but at the same time more confidence in the intuition-based **output** (our expert judgment). How is this conflict solved?
  - A strongly analytical person: Trust the model
  - A strongly intuitive person: Trust the intuition
  - Conflict-averse person: Adjust the model input so that it gives the desired output. In the example, this may be achieved through categorization of some of the medium complex user stories as “simple”. This conflict-avoiding adjustment may happen both consciously and unconsciously.

Experts can be very good, BUT ...

- are frequently outperformed by simple models
  - E.g., in many types of clinical judgment and effort estimation uncertainty judgments
- can be extremely inconsistent
  - E.g., our studies on expert estimation of software development effort
- may be unable to transfer extensive knowledge into accurate judgment
  - E.g., mutual funds
- are impacted by many irrelevant factors
  - E.g., the weather may impact how people's abilities are judged (see next page)

Example of impact from irrelevant information

- We divided 65 software professionals randomly into three groups: Low (22 participants), Control (23 participants), and High (20 participants).
- We gave all participants the same programming task specification but varied the words describing some of the requirements slightly.
- The most notable difference in wording is that we asked the:
  - Low group to complete a “minor extension”
  - Control group to complete an “extension”
  - High group to develop “new functionality.”
- We told all the estimators:
  - “You shouldn’t assess how much the client will spend on this project, but what’s required by development work with normal delivery quality.”

Priming II - results

- The resulting median effort estimates were
  - Low (minor extension): 40 work-hours
  - Control: 50 work-hours
  - High (new functionality): 80 work-hours

So, when should we trust experts?

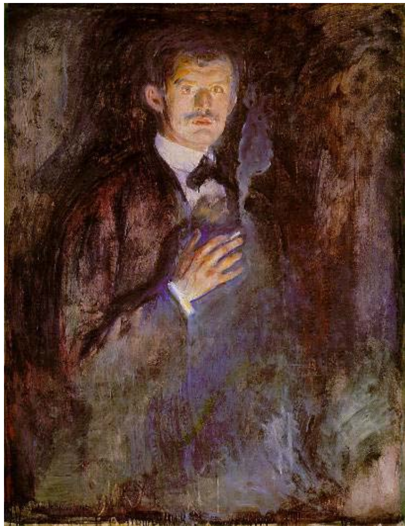
- When they have extensive "deliberate practice" in the particular problem to be solved.
  - See studies by Ericsson and by Shanteau.
- When the context includes little irrelevant and/or misleading information leading to well-known effects (dilution, anchoring, priming, wishful thinking).
  - See the "human biases" studies, e.g., by Kahneman & Tversky

Indicators of estimation expertise

- Length of experience?
  - Not a good indicator.
- Experience from similar projects?
  - Definitively yes, but remember that expertise is “narrower” than typically assumed.
- The best developer?
  - Not always. The best developer may not be suited for the estimation of work effort for novices.
  - “Outside view” (less know-how) sometimes a better strategy.

Indicators of estimation expertise

- The one with highest confidence in his/her estimate?
  - No. We observed the opposite. The most confident are typically the most over-optimistic.
- Those historically most accurate?
  - Yes, but not a very good indicator. We observed that the software professional (out of two) most over-optimistic on previous estimate had a 70% probability of being the most over-optimistic on the next estimate.
- Personality? (optimism tests, suggestibility, Big five test, IQ-test, ...)
  - Probably not of much help.
- Slightly depressive people?
  - Yes ☺. They are on average most realistic regarding own abilities.



Effort estimation uncertainty analysis

Probabilities: A late invention (and we are not good at assessing it)

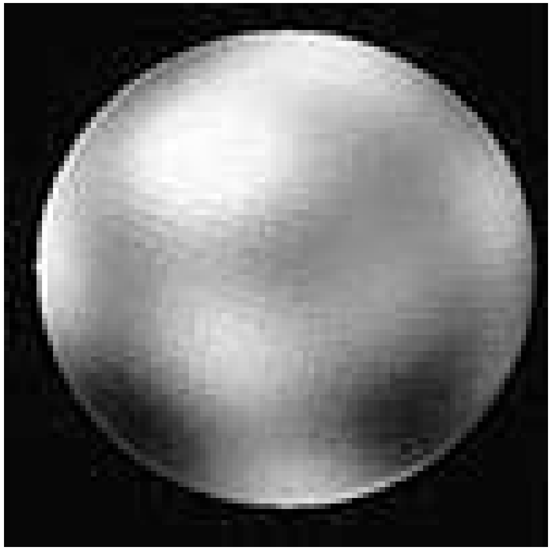
Task: What is the number of inhabitants in Norway

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”?

- 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”

Realism-conflicting goals

- Informative assessments excludes wide (realistic) intervals
- Rewards for over-confidence
  - Realism used as indicator for lack of skill!
- The clients don't like high uncertainty ….
- If the uncertainty is too high we will not be allowed to start this project ….

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 ...

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.