

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

Steria, June 2012

Magne Jørgensen

Simula Research Laboratory & 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: <http://simula.no/people/magnej/bibliography>
  - 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.

## Estimation error

- Average estimation overrun in IT-projects is reported to be about 30%
  - Large estimation error sometimes causes huge problems with project management, profitability, client satisfaction and investment analysis!
  - No substantial changes in average estimation error from 1970 until today.
  - Seems to be similar levels of estimation errors and biases in all cultures.

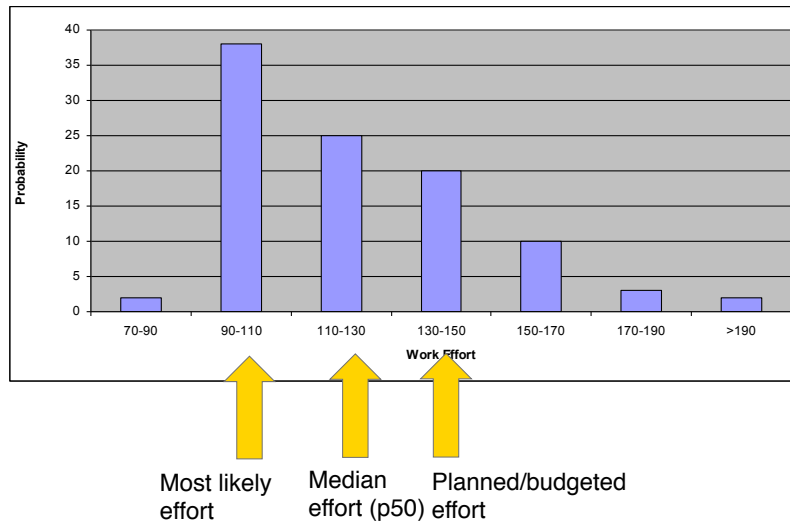
3

**Do we know what we mean by "estimate"?**



4

**We benefit from thinking probabilistically about effort usage (especially to enable good communication about what we mean by an effort estimate!)**



5

## 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 p70%-estimate). Purpose: **Project control**.
  - Budget (e.g., based on an p80%-estimate). Purpose: **Financial control** of project portfolio.
  - Price (e.g., based on p40%-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!

6

## Indicators of high risk of effort and cost overrun



## Indicators of projects likely to overrun their estimates

- Factors we may have to accept (or stop doing complex projects):
  - We do things that are substantially different from what we have done before
  - There are many interfaces to other systems and/or many stakeholders
  - A substantial re-engineering of existing work processes is involved
  - The problems to be solved are complex
  - Bad luck (could be many small "bad lucks" or one large)
- But, there are factors where we can and should improve:
  - Ambition level
  - Situational and human biases
  - Competence of client and provider
  - Attention, supervision and management support.
  - Communication with providers, sub-contractors, clients and other stakeholders, including cultural issues.
  - Bidding processes (avoiding winner's curse, adverse selection, ...)
  - Development methods.

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



The better-than-average effect....



## Over-confidence ...



11

## Wishful thinking

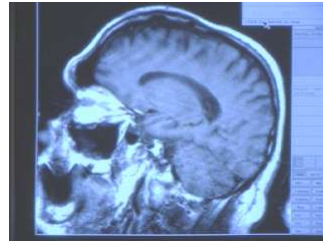
- Mix of “I hope this does not take more than ...”
- “To be a good programmer I should not use more than ...”
- Optimism and over-confidence can lead to increased performance, BUT
  - Only for a short period of time.
  - The effect is over-rated.



12

## 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?



13

## 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

14

## Winner's Curse

- One overlooked reason for cost overrun seems to be the so-called "winner's curse".
- Winner's curse (WC):
  - Adverse (biased) selection of providers and/or projects wrt over-optimism:
    - Software projects tend to be won by providers with over-optimistic bids.
  - It does not help much being realistic in 9 out of 10 bids, if the only bidding round won is when being over-optimistic.
  - WC leads to lower than expected profitability of software providers.

15

## Our Findings

- Empirical study on bidding
  - Winner's curse => Client's curse.
    - WC => More work completed by the clients.
    - WC => Lower quality.
    - WC => Delays.
  - Low price indicator of low expertise.
  - An over-optimistic price may make even experienced and potentially very good providers poor.

16



## Our Findings

- The factors determining the harmful effect of WC include:
  - Price focus of the client
  - Number of bidders
  - Uncertainty of cost estimates
  - Awareness of the WC-effects of the bidders and the client
  - Degree of opportunistic behavior
- **Recommendation:** Do the "thought-experiment": What if our bid is the lowest among ten others and we get selected. How likely is it that we will make a profit?

17

## Other implications

- Cost estimation accuracy surveys are not based on a random sample of all project estimates.
  - Reported average cost estimation accuracy should therefore NOT be interpreted (and described) as a measure of the estimation ability of the software industry!

18

## Expert estimation



### 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.

## **An empirical study**

- 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.”

21

## **An empirical study - results**

- The resulting average (median) effort estimates of the different groups were:
  - “Minor extension” group: 40 work-hours
  - Control group: 50 work-hours
  - “New functionality” group: 80 work-hours

22

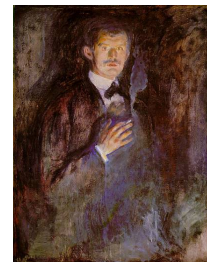
## Indicators of estimation expertise

- Length of experience? **Not a good indicator.**
- Experience from similar projects?
  - Definitely 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.
- The lowest bid? **No! Perhaps the worst indicator.**

23

## Indicators of estimation expertise

- The one with highest confidence in his/her estimate?
  - Perhaps, but we have also observed the opposite. The most confident may also be 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.



## Recommendations

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.

25

## Hvor mange non-stop (alle farger)?

