[simula . research laboratory]

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

Mexico, August 2011

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*
  - 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.

2

A few words about the study

- Probability ignorance of min-max intervals
- Assimilation effects
- Sequence effects
- Relative estimation

- Relevance to real-world project estimation?

3

Estimation error

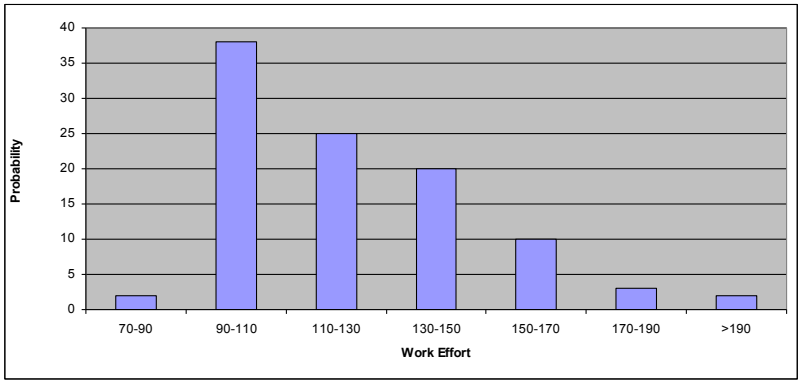
- Average estimation overrun in IT-projects is reported to be about 30%
  - Sometimes the estimation error is 200% and more.
  - Large estimation error sometimes causes huge project management, profitability, client satisfaction and investment analysis problems!
  - No substantial changes in average estimation error from 1970 until today. Why cannot we learn from previous experience?

4

Do we know what we mean by ”estimate”?

5

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



6

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!

7

[simula . research laboratory]

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

UNIVERSITY OF OSLO

The better-than-average effect….

9

Over-confidence …

10

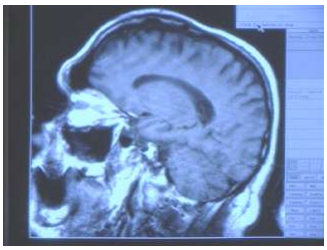
Wishful thinking

- 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.

11

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?



12

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

13

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.

14

[simula . research laboratory]

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.

16

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."

17

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

18

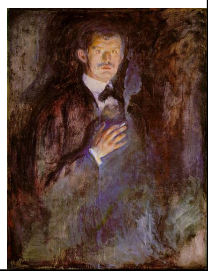
Indicators of estimation expertise

- 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.
- The lowest bid? **No! Perhaps the worst indicator.**

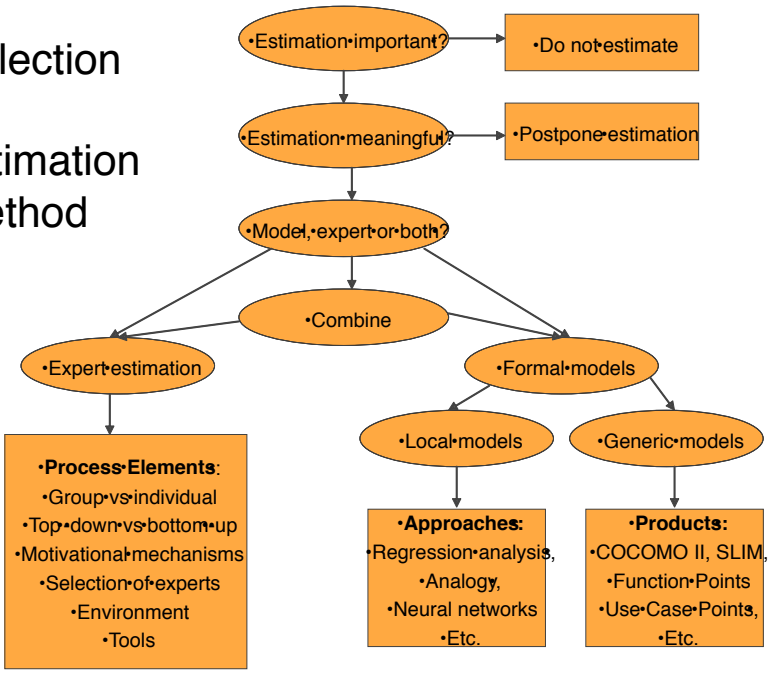
19

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.



Selection of estimation method



[simula . research laboratory]

Effort estimation uncertainty analysis

UNIVERSITY OF OSLO

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

23

Task: What is the number of inhabitants in Norway

Minimum

Maximum

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

24

How sure is "almost sure"?

- Our field studies of software companies found that:
  - Almost certainty, this mean about 60% certain
  - "60% certain" = "75% certain" = "90% certain = "99% certain"

25

Why are the intervals too narrow?

- 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!

26

Two views on assessing 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".

27

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.

28

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.

29

A good uncertainty assessment 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.

30

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?

31

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.

32