[simula . research laboratory]

Software Development Effort Estimation: What do We Know About It?

PGS, Wroclaw, Nov. 2009

Prof. 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: *simula.no/research/engineering/projects/best*
  - Extensive industrial experience as programmer, project manager, process improvement managers and general manager.
- Conduct advisory work and seminars for software companies.

2

Last time I was here ….

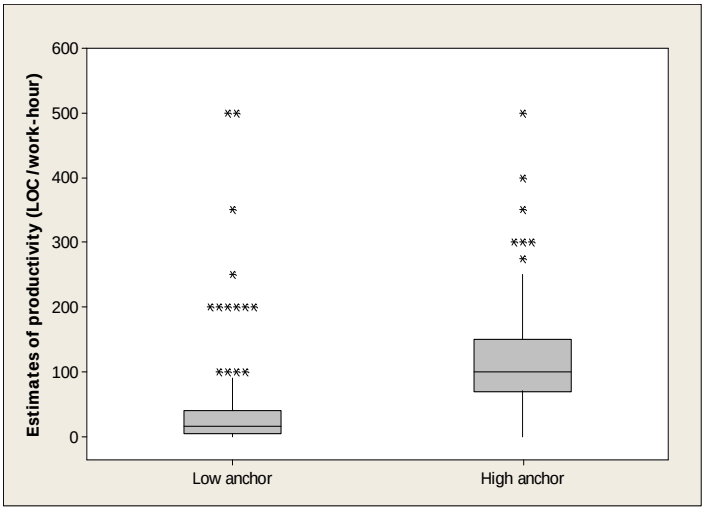
- Many studies in one …
- Main topics:
  - Anchoring effects
  - Cultural effects
- Participants: 373 developers from Romania, Ukraine, Polen, Nepal, India and Vietnam

3

Estimation Task 1

- **Low Anchor Group**: Did you on average write more or less than 1 Line of Code per work-hours in your last project?
- **High Anchor Group**: Did you on average write more or less than 200 Lines of Code per work-hours in your last project?
- Then, on the next page, the developers from both groups were asked to estimate the number of lines of code they wrote in their last project.

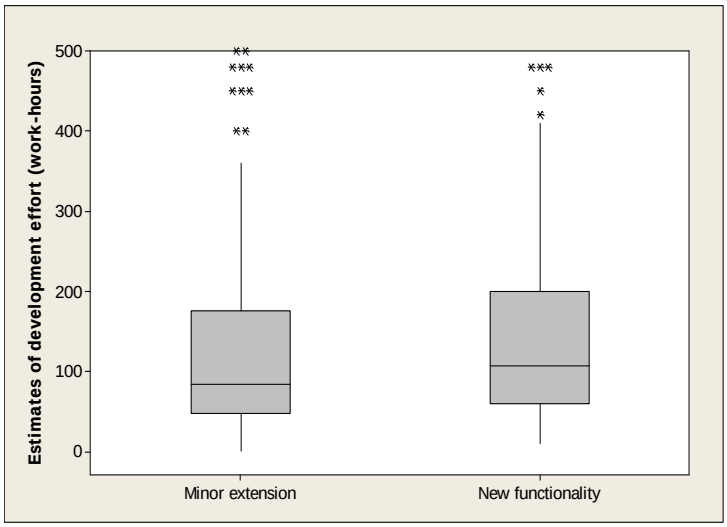
Results



Estimation Task 2

- One group received the description that the development task to be estimated was a “minor extension” and one group the description that the task to be estimated constituted “new functionality”.
- The development task to be estimated and the rest of the instructions were exactly the same.

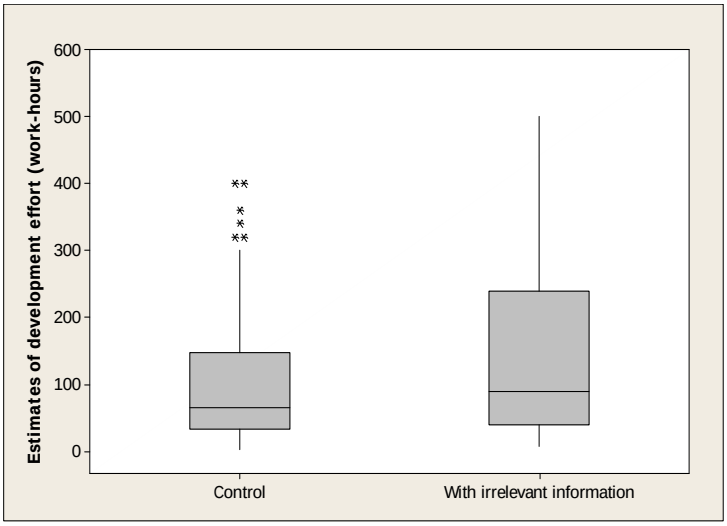
Results



Estimation Task 3

- One group received a specification of software with mainly effort-relevant information and one group that received the same effort-relevant information, but in addition received much information that had no intended relevance for the development effort.
- The actual irrelevance of the information for the purpose of effort estimation was confirmed by an independent, experienced software developer.

Results



Differences between the countries?

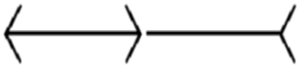
	Estimation Task 1			Estimation Task 2			Estimation Task 3		
Group	**Low anch.**	**High anch.**	**Diff**	**"Minor ext."**	**"New func."**	**Diff**	**Control**	**Irr. Inf.**	**Diff**
India	25	150	125***	63	80	17	30	58	28*
Nepal	11	120	109***	50	152	102*	80	90	10
Poland	12	100	88***	102	110	8	80	100	20
Romania	10	70	60***	95	100	5	50	70	20
Ukraine	10	100	90***	120	120	0	60	200	140*
Vietnam	25	100	75***	90	120	30	100	100	0

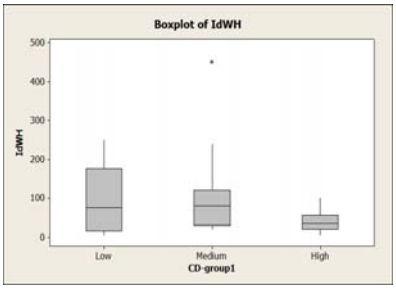
Level of ”interdependence” may be important

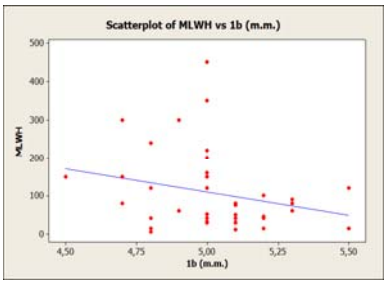
Instrument		Estimation Task 1			Estimation Task 2			Estimation Task 3		
		Low anch.	High anch.	Diff.	Min. ext.	New func.	Diff.	Contr.	Irr. inf.	Diff.
Interdependence	High	15	100	85***	70	100	30**	50	100	50**
	Low	15	105	85***	120	120	0	80	80	0

Higher interdependence (higher context-dependency)
➔ Lower estimates and easier biased

Same phenomenon found on impact from optical illusions?

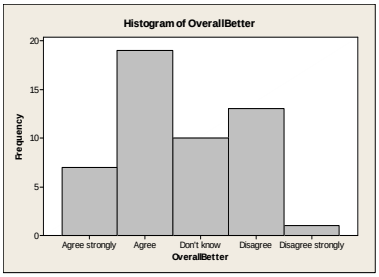




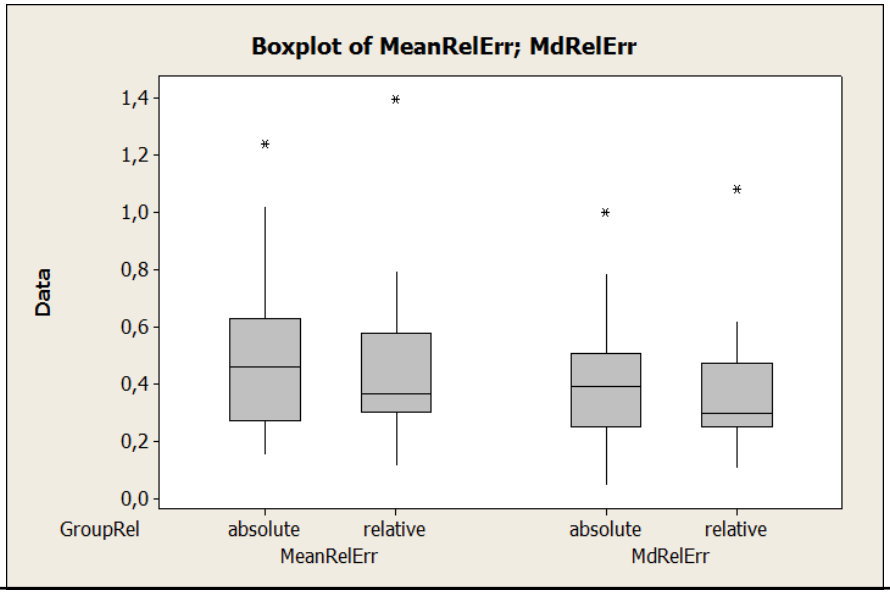


Are Agile Methods Better?

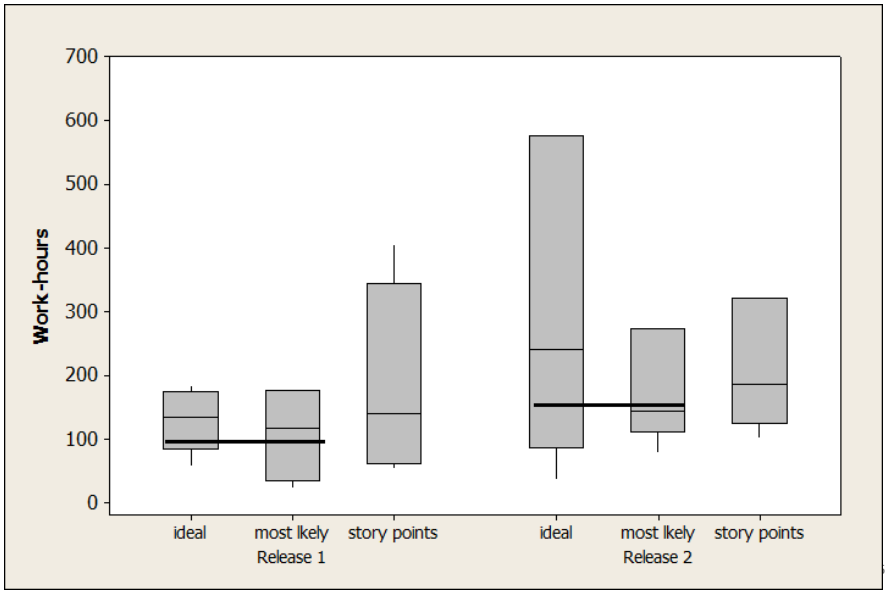
- **Question**: How much do you agree in: *"Use of agile methods has caused a better performance when looking at the combination of productivity and user satisfaction."*
- **Study**: Presentation of **randomly generated** data sets.
- **Result**: Bias in favor of agile methods (see figure).
  - The agreement in the claim depended on previous belief in agile methods.
  - Previous belief: Agile methods are better ➔ 20 of 32 agreed
  - Previous belief: Agile methods are not better ➔ 1 of 7 agreed
  - Previous belief: Neutral ➔ neutral answers
- The real-life bias is probably much stronger:
  - Lack of objective measurement. More bias in favor of the preferred method.
  - More variables of importance, i.e., more complex interpretation and more space for wishful interpretation.



This Study (warm-up tasks)



This study (User Story estimation)



[simula . research laboratory]

What We Know About Effort Estimation

(and how to improve it)

A very brief introduction …

UNIVERSITY OF OSLO

16

80% of software developers are “better than average”, only 2% worse

17

“Almost sure” (99% or 90% confident) of being inside a min-max interval corresponds to a “hit rate” of about 70% …

18

Task: What is the number of inhabitants in Norway

Minimum

Maximum

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

19

Confusing Estimation Terminology

Is the estimate: 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, something else?

20

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") can be downloaded from our website.

21

Characteristics of the Estimation Error

- Most large scale surveys of software projects finds an average estimation overrun (over-optimism) of about 30%.
- No cultural differences.
- No improvement over time.
- Small tasks are over-estimated, large tasks under-estimated (Vierordt's law)
- Unexpected or forgotten activities is the main error reason in large projects.
- High level of inconsistency.
- No improvement from use of formal estimation models, such as COCOMO and Function Points.

22

BUT, 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 meaningful.
- 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.
- **Question**: It turns out not to be a good idea to give project managers bonuses based on low estimation error. Why not?

23

[simula . research laboratory]

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

UNIVERSITY OF OSLO

Motivation

- Strong connection between high motivation for low use of effort (“commitment”) and over-optimism
- Optimism can have a positive impact on performance, BUT
  - Only for a short period of time in the beginning.
  - It’s easy to over-evaluate the effect.

25

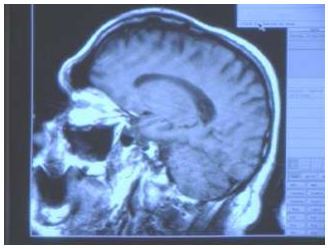
Motivation (cognitive dissonance)

- 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.

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?
- Over-optimisms increases with "psychological distance"



27

Bidding round format frequently leads to over-optimism

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

28

Ten Recommendations:

1. Educate a ”cost engineer” responsible for checklists and collection of experience/historical data.
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 when estimating and assessing uncertainty
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 improved 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.