[ simula . research laboratory ]

# How should we (not) design empirical studies of software development?

Magne Jørgensen & Stein Grimstad
Simula Research Laboratory
Oslo, Norway
magnej@simula.no

By thinking constantly about it ...

## The Software Engineering Department at Simula Research Laboratory

| | Employees |
|---|---|
| 1 | Research director (Dag Sjøberg) |
| 7 | Researchers (Lionel Briand, Erik Arisholm, Magne Jørgensen, Stein Grimstad, Bente Anda, Amela Karahasanovic, Jo Hannay) |
| 7 | PhD students (James Dzidek, Vigdis By Kampenes, Tanja Gruschke, Gunnar Bergersen, Hans Christian Benestad, Nina Elisabeth Holt, Kristin Børte) |

Three (four) groups within the SE-department:

**- Estimation of Software Development Effort**

- Object-Oriented Analysis and Design
- Software Engineering Research Methods
- (Testing)

## The estimation group

- **Main goal**: Improvement of judgment-based software development effort and uncertainty estimation processes
- **Research topics related to effort estimation**:
  - Judgmental forecasting
    - Understanding the ”magic step” of expert estimation
  - Overconfidence / overoptimism
  - Learning / training processes
  - Bidding processes
  - Effort estimation in agile software development projects



## My goal with this presentation

- Start a discussion and get your experience and opinions on how to conduct empirical software engineering studies
- For this purpose I will:
  - Exemplify how we do it at Simula Research Laboratory
  - Present three viewpoints on shortcomings on current empirical software engineering research.

## Can you guess what my example will be about?



## Number of inhabitants in Norway in 2001

Minimum


Maximum

Be 99% confident to include the true value!

## Radius of the (dwarf) planet Pluto

Minimum

Maximum

Be 99% confident!



## Example: Overconfidence in the accuracy of effort estimates

- **Step 1**: Motivate research topic and review existing work
  - Overconfidence regarding accuracy of estimates leads to poor plans, budgets and investment analyses.
  - Inherent uncertainty in effort estimates:
    - "*Anyone who expects a quick and easy solution to the multi-faceted problem of resource estimation is going to be disappointed. The reason is clear: computer program system development is a complex process; the process itself is poorly understood by its practitioners; the phases and functions which comprise the process are influenced by dozens of ill-defined variables; most of the activities within the process are still primarily human rather than mechanical, and therefore prone to all the subjective factors which affect human performance.*" (Alfred M. Pietrasanta at IBM Systems Research Institute in 1968.)
  - Few studies on this topic.
  - Review of related work in psychology, forecasting, cognitive science, software engineering, ….

## Overconfidence in the accuracy of effort estimates

- **Step 2**: Understand the topic (preferably, in the field)
  - Our field studies of the state-of-practice suggested that:
    - Most common: No formal assessment of uncertainty.
    - Sometimes: Minimum-maximum values without any indication of confidence level.
    - Very few: Prediction intervals (interval + confidence), e.g., "90% confident in including actual effort in the interval [1000; 1500] work-hours.
  - Field data suggested that "almost certain" means "60-70% probable" and that:
    - Narrow, overconfident prediction intervals are rewarded
    - The uncertainty assessment learning environment is "wicked"
      - Poor feedback, on a format difficult to learn from.

➔ Frustrated clients, poor project control.



## Overconfidence in the accuracy of effort estimates

**Step 3**: Experiment to understand the underlying cause-effects.

**Example**: Experiment on interpretation of confidence level …
**Main result**: 50%, 75%, 90% or 99% confidence made no difference!
**Possible implications**:

1) The PERT method (typically recommended in textbooks) will not work well?
2) Meaningless to ask people to be "90% confident" without training/support?
3) We need to change how we elicit uncertainty or how we train people?

**Results**: The results are described in Table 3.

| Group | Confidence Level | Number of Students | Median PIWidth |
|---|---|---|---|
| 1 | 50% | 28 | 0.8 |
| 2 | 75% | 26 | 0.7 |
| 3 | 90% | 24 | 0.7 |
| 4 | 99% | 27 | 0.7 |

**Table 3: Prediction intervals for estimates based on different confidence levels**

## Overconfidence in the accuracy of effort estimates

- **Step 4**: Experiment with alternative processes
  - Test different elicitation formats:
    - Traditional approach (PERT): "What is the maximum effort? (Be 98% sure)"
    - Alternative 1: "How likely is it that the effort overrun will be more than 50%?"
    - Alternative 2: "What is the maximum effort? How likely is it that the actual effort will be higher?"
    - Main finding: Alternatives 1 and 2 were better than the traditional approach (rewrite project management textbooks?)
  - Test different ways of producing and presenting historical data
    - Main findings:
      - Easier to support Alternative 1 and 2 with historical data than the traditional approach.
      - Even when historical data is produced and assessed to be relevant by the estimator, they do not always use them
        - » BUT, on average, it helps…



## Overconfidence in the accuracy of effort estimates

- **Step 5**: Evaluate the approaches in the field
  - Real-life randomized controlled trials showed the same positive effect of using Alternative 1 in comparison to the PERT approach.
  - Research method:
    - A company randomly allocated the three alternatives to uncertainty assessments of software projects.
    - We paid them for the extra work (actual work on the uncertainty assessment + administration)
  - Convincing results. But, you should never trust studies (this time, our own) that show favorable results for own "inventions". Hopefully, other researchers will soon try to replicate/falsify my results.

## Overconfidence in the accuracy of effort estimates

- **Step 6**: Ongoing work …
  - Better training and feedback approaches
    - Strange results: “Performance review” improved efficiency, made estimation accuracy worse, and left uncertainty assessment realism unchanged.
  - Better uncertainty elicitation methods
    - Perhaps a combination of model and judgment
  - Better understanding of the mental processes involved
    - Irrelevant information impacts …



## These 6 steps are hardly controversial …

- The general process illustrated in the example is not original. It is well within the ”Hypothetico-deductive model”.
  - I would be surprised if you disagreed with a single element of the research process outlined.
- The interesting issues (if any) are related to HOW we conduct the studies and, perhaps, some of the underlying ideas regarding research artificiality and realism.

## Issue 1

- We conduct a high number of small experiments on software professionals integrated in presentations at industry conferences and seminars:
  - Experiments related to presented topic
  - Results (not all) presented in the end of the presentation
  - Since 2004: approx. 30 such experiments per year
- **Experience**: Highly valuable research results. High interest in our seminars, i.e., many experienced software professionals as participants. Low cost. Improves the presentations.
  - Several of the companies wants to replicate the experiments on future seminars, i.e., we get replication of the experiments conducted by the companies themselves!
- **Observation**: We are not aware of any other SE research groups that uses this industry conference and seminar experiment opportunity extensively. What about you?



## Issue 2

- Many of our experimental designs are deliberately artificial to, e.g.,:
  - Isolate basic mechanisms
  - Demonstrate the existence of phenomena
  - Demonstrate the persistence of phenomena.
- **Observation**: A review of software engineering experiments published from 1993 to 2002 gave that artificiality (e.g., small tasks in place of more industry alike tasks) was nearly always seen as a threat to external validity and hardly ever as a means to generalize.
  - Generalization from sample to population was the dominating approach, which is strange given how meanlingless this approach frequently is. How able are we, for example, to define the population of SE tasks and contexts?
  - We typically need to base valid generalizations on theory and a variety of different types of argumentations, not so much on inferential statistics.

## Issue 3

- A few of our experiments are very expensive (> 100 000 US dollars).
  - We have the flexibility to conduct such experiments instead of hiring more research staff.
  - We apply for (and receive) research grants for this purpose.
- **Experience**: This enable better experiments.
  - We are, for example, able to hire software professionals as participants in experiments that last several days.
  - We pay companies for their extra effort related to logging of more information and test of new processes.
- **Observation**: Other SE groups seems to have strong budget limitations on their studies. While it is accepted to buy expensive hardware, it is not accepted to spend much money on SE experiments. Why is this so? Should it be changed?



## Other: Themes from the abstract …

I will summarize experiences from empirical studies conducted at the Software Engineering Department at Simula Research Laboratory. The experiences are illustrated with study design and results from my own research on software cost estimation. Topics that I would like to present and discuss include: 1) What should we learn from other disciplines' research methods and results? 2) When is artificiality in experiments a threat to validity? 3) How should we conduct studies with high degree of realism? and 4) What shall we do with the immature and misleading use of statistical hypothesis testing in software development research?

## “Clouds Make Nerds Look Better”

- Study of university applicants:
  - Nerds had significantly higher chance compared to non-nerds on cloudy days.
    - Nerd-factor measured as academic rating divided by social rating (e.g., leadership).
  - 12% higher chance when sunshine compared to worst cloudcover.



## Irrelevant and misleading information in requirement specifications …

- There are good (and not so good) reasons for this, e.g.,
  - Information added for pricing purposes, or other purposes than effort estimation,
  - ”copy-paste” of general information about the clients’ processes and organization from previous specifications, and,
  - lack of compentence in how to write a good requirement specification.

## The impact of the # of pages

- Computer science students estimated the effort of the same programming task.
  - Group A: Received the original specification, which was one page long.
  - Group B: Received a version of the specification that had exactly the same text, but was seven pages long. The increased length was achieved through double line space, wide margins, larger font size and more space between paragraphs.
- Group A and B's estimates were, on average, 117 and 173 work-hours, respectively.
  - Longer specification ➔ higher estimates.



## Adding irrelevant information …

- Group A software professionals received the original programming task specification.
- Group B software professionals received the same specification, with clearly estimation irrelevant information included.
- Results:
  - Group A average: 20 work-hours
  - Group B average: 39 work-hours

## Misleading information …

- HIGH (LOW) group: *"The customer has indicated that he believes that **1000 (50)** work-hours is a reasonable effort estimate for the specified system. However, the customer knows very little about the implications of his specification on the development effort and you shall not let the customer's expectations impact your estimate. Your task is to provide a realistic effort estimate of a system that meets the requirements specification and has a sufficient quality."*
- *Results:*
  - *HIGH anchor group average: 555 work-hours*
  - *CONTROL group (no anchor) average: 456 work-hours*
  - *LOW anchor group average: 99 work-hours*
- *None felt they had been much impacted, and most of the software professionals claimed that they had not been impacted at all.*



## Impact from "future opportunities"

- Group-WISHFUL:
  - "[the client] *has invited many providers (more than 10) to implement these extensions and will use the providers' efficiency on this project as important input in the selection of a provider for the development of the new ticketing system … Estimate the work effort you think you MOST LIKELY will use to complete the described extension to the existing ticketing system. The estimate will not be presented to* [the client] *and should be the effort you most likely will need.*
- *Results:*
  - *Group-WISHFUL: 40 work-hours*
  - *Group-CONTROL: 100 work-hours*

## Why does this happen?

- Hot topic among researchers. We do not know much.
- The main reason is that brain activity when estimating effort is mainly unconscious, i.e., we are not in control of most of our thought processes and attention.



## Example: The Cocktail Party Effect

## HELP! My brain is out of control …

- The lack of brain control implies that it is hard to defend positions like:
  - "*I know why I like what I like*"
  - "*My expert judgment-based estimate is based on information X, by not by information Y.*"
  - "*I will not be impacted in my judgment by a dinner with a potential provider*"
- This is, however, what most people seem to do.
- The reason for our unwillingness to accept the lack of control may be a strong desire to believe that we are rational individuals.
  - The rational reaction to our lack of control is to admit irrationality and learn how to live with and avoid it.



## We cannot be that irrational, or we would have been extinct …

- The effect of irrelevant information is a consequence of high performance tailored (evolved) to other, much more important, situations (survival and reproduction) combined with the relatively slow speed of mental activities:
  - Information received: ~ 10 Mbit/sec
  - Information processed consciously (working memory): ~ 40 "bit"/sec?
- If the working memory (the conscious part of our brain) should do all processing work, we would not be able to walk and talk at the same time - probably not even walk or talk.

## There are individual differences …

Evidence suggest that people more affected by irrelevant and misleading information have:

- Poorer memory.
- A higher disposition towards absorption.
- Higher level of depression.
- Stronger emotional self-focus.
- A tendency to be more easily bored.
- A more external locus of control.
- Better imagery vividness.



## Theory: Threshold of belief updating …

- An underlying theory for stronger impact from information is the a theory related to connection between the brain hemispheres.
  - Possibly related to differences in size of *corpus callosum* and activation of right brain hemisphere.
- Handedness may be a (far from perfect) measure of the belief updating threshold.
  - Supported by several of our studies in software engineering contexts.
- Example: Anchoring experiment on estimated time to read and answer mail the following day:

**Table 1: Median Estimated Time**

| Group | Mixed-handed | Strong-handed |
|---|---|---|
| **A (5 min)** | 7,5 min (n=12) | 10 min (n=12) |
| **B (4 h)** | 15 min (n=17) | 10 min (n=8) |
| **C (10 h)** | 15 min (n=13) | 10 min (n=8) |
| **D (22 h)** | 30 min (n=14) | 7,5 min (n=8) |

## Is it better to have a high or low update threshold?

It depends ….

- We need a certain degree of stability in our beliefs (consistency) to benefit from our past experience.
  - There is a substantial lack of consistency in software development effort estimation.
  - In an experiment with software professionals Stein Grimstad found that the mean difference of the effort estimates **of the same task by the same estimator** on different occasions was as much as 71%.
- We also need a certain degree of flexibility (belief updating) to benefit from new experiences.
  - Mixed-handers (lower threshold for updating?) had systematically higher (and more realistic?) software development effort estimates when there were no irrelevant information.
- There is consequently a fine balance between being consistent (and less impacted from irrelevant and misleading information) and being flexible.



## What we definitely should avoid …

- Exposure to obviously irrelevant information, e.g., customer expectations that will have the role as anchors in effort estimation situations.
- A belief that the impact from irrelevant information only happens to other than yourself.
  - This will effectively prevent actions to take place.
- Information that “dilutes” the impact from the most essential information.
  - Much evidence to support the claim that more information of lesser quality or relevance typically leads to too little emphasis on the most relevant information.

## Example: The “dilution” effect

Software professionals were asked to weight the importance of estimation model selection factors. A 20% weight meant, for example, that the score of of model on that factor would count 20% of the evaluation. The sum should be 100%.

**The factors were:**

1. Accuracy of the estimates
2. Ease of understanding the model
3. Ease in use of the model
4. The model uses only easy available data
5. The method is flexible and possible to use when not all input data are available
6. The method provides minimum-maximum intervals
7. Other factors

**Group A** had a reduced list of factors (Factors 1-3 + 7), while **Group B** had all seven factors.

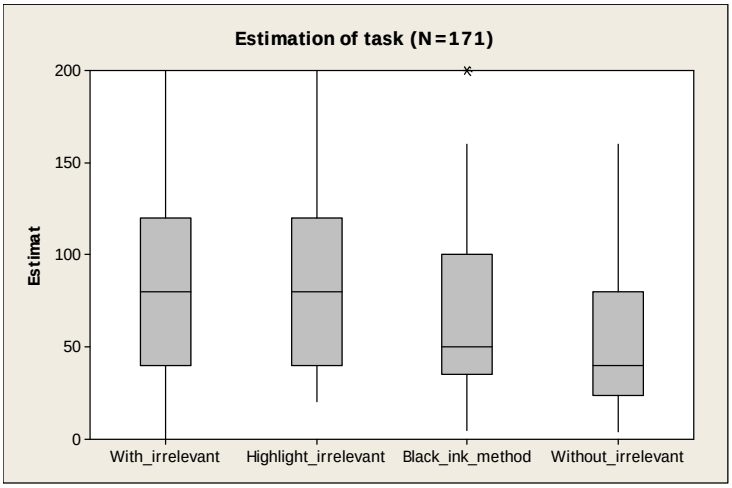
**The most important factor (Factor 1) had the weight 40% in Group A, while ”diluted” to only 24% in Group B.**



## Debiasing techniques …

- Awareness of own biases does not help directly, but indirectly in that other less vulnerable judgment processes are chosen.
- Analytic, as opposed to intuition-based, estimation processes may help, e.g., estimation models and use of historical data.
  - But, as long as they are not mechanical, there is room for impact from irrelevant and misleading information.
- The “black-ink method” (see next slide) may help.
- The only really effective method is to remove the irrelevant and misleading information. Our recommendation is based on this finding.
  - Debiasing techniques are typically the second best option.

## The “Black Ink”-method (experiment at JavaZone 2006)






## What to Do? Recommendations

### Element 1: Filter Estimation Information

- People other than those estimating the effort should prepare a filtered estimation information package that includes relevant and neutral estimation information only.

## What to Do? Recommendations

**Element 2: Less Vulnerable Estimation Work**

- Exclude estimators that deliberately or accidentally gain access to misleading or irrelevant information that can bias the estimates. In particular, the estimator should not know the "desired" outcome of the estimation process, because this probably will induce wishful thinking.
- Exclude estimators with vested interests in the outcome of the estimation process, e.g., estimators that are very keen on starting the project and may easily fall prey to wishful thinking.



## What to Do? Recommendations

**Element 3: Adjustments**

- When the estimation work has been completed, there may be a need for adjustments and re-estimation.
- This situation may be highly vulnerable to wishful thinking and should be treated very carefully.
- We recommend that the software professional in charge of producing the filtered estimation information package, and not the estimators, updates the information to include less functionality, lower quality, simplified design, or apply other means of reducing the required effort.
- The estimators should then be asked to re-estimate the effort based on the updated estimation information package. Under no circumstances should the estimators know the desired outcome or receive information that suggests that they need to estimate more optimistically.