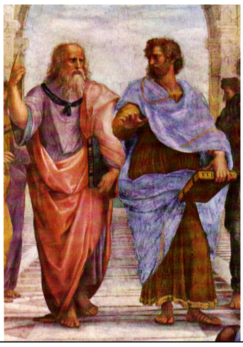
[simula . research laboratory]

From myths, fashion, over-simplifications and non-validated claims to evidence-based software engineering

Magne Jørgensen
Simula Research Laboratory
&
University of Oslo, Norway



Most of the methods below have once been (some still are) fashionable ...

- The Waterfall model, the sashimi model, agile development, rapid application development (RAD), unified process (UP), lean development, modified waterfall model, spiral model development, iterative and incremental development, evolutionary development (EVO), feature driven development (FDD), design to cost, 4 cycle of control (4CC) framework, design to tools, re-used based development, rapid prototyping, timebox development, joint application development (JAD), adaptive software development, dynamic systems development method (DSDM), extreme programming (XP), pragmatic programming, scrum, test driven development (TDD), model-driven development, agile unified process, behavior driven development, code and fix, design driven development, V-model-based development, solution delivery, cleanroom development, ...
- Did we go from RUP (previous fashion) to agile (current fashion) because of that was what the evidence told us?

Where I think we should go ....

- Choices of software engineering practices should be based on evidence on what works.
  - Evidence from research, but also systematic collection of practice-based experience and evidence from "own studies" are valid.
- We should improve the ability among software engineers to collect, evaluate and synthesize evidence.

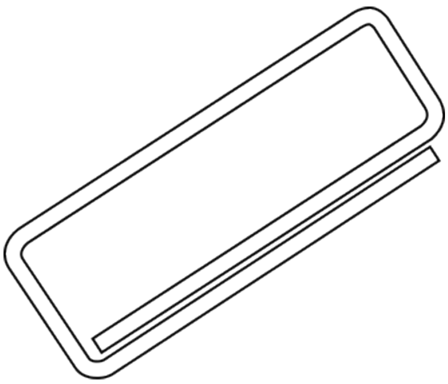
What to do: Evidence-based software engineering (EBSE)

- The main steps of EBSE are as follows:
  1. Convert a relevant problem or need for information into an answerable question.
  2. Search the literature and practice-based experience for the best available evidence to answer the question.
  3. Critically appraise the evidence for its validity, impact, and applicability.
  4. Integrate the appraised evidence with practical experience and the client's values and circumstances to make decisions about practice.
  5. Evaluate performance in comparison with previous performance and seek ways to improve it.

Tore Dybå, Barbara Kitchenham and Magne Jørgensen, Evidence-based Software Engineering for Practitioners, IEEE Software, Vol. 22, No. 1, Jan-Feb 2005.

WHY DO WE BELIEVE IN MYTHS?

[simula . research laboratory]

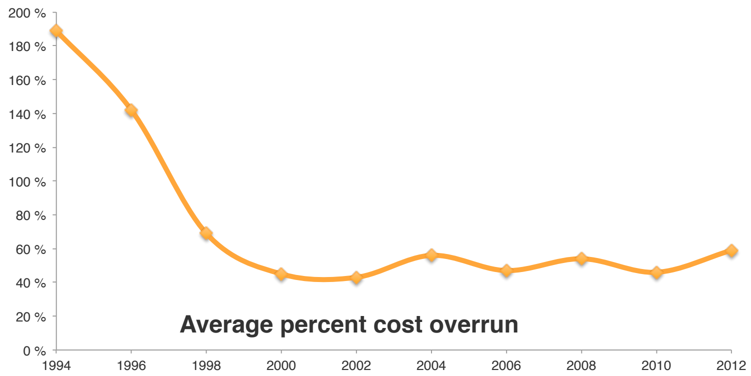


[simula . research laboratory]

[simula . research laboratory]

[simula . research laboratory]

Is there a ”software crisis”?



THE STANDISH GROUP

(page 13 of their 1994-report): “*We then called and mailed a number of confidential surveys to a random sample of top IT executives, asking them to share failure stories.*”

Difficult to remove myths ...

How large are software cost overruns? A review of the 1994 **CHAOS report**
M Jørgensen, K Moløkken-Østvold - Information and Software Technology, 2006 - Elsevier
The Standish Group reported in their 1994 **CHAOS report** that the average cost overrun of software projects was as high as 189%. This figure for cost overrun is referred to frequently by scientific researchers, software process improvement consultants, and government **...**
Sitert av 153 Beslektede artikler Alle 11 versjoner Sitér Lagre

Years of critique may have had an effect (from the CHAOS Report – 2013):

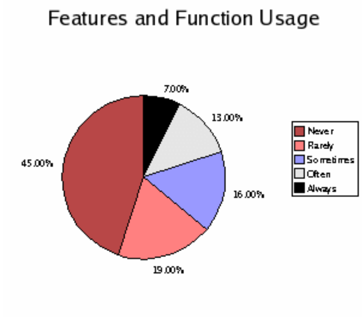
90,000 completed IT projects. However, for our new database we eliminated cases from 1994 though 2002, since they did not match the current requirements for analysis. The new database has just under 50,000 projects.

The Standish Group - again

Claim: 45% of the functionality (when using ”traditional” development methods) are never used

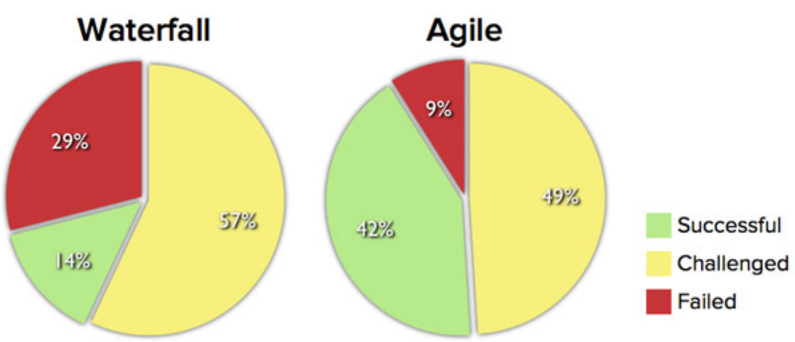
– Reference to a keynote by Jim Johnson at XP 2002. Several requests about the method and the results, but no answers. No documentation available.

- What is the meaning of 45% of functionality never used? (Never by anyone? By the person asked? Who did they ask?)
- Is this for one particular system or a survey of many systems?
- Is there reasons to believe that the use of traditional method is the **reason** for the low use of functionality?



[simula . research laboratory]

... and again



Source: The CHAOS Manifesto, The Standish Group, 2012.

Success = “On time, on schedule and with specified functionality”

Can you find one serious error of this study?

[simula . research laboratory]

Over-simplifications

Claim: The cost of changes (or correction of errors) increases with a factor of 10 with each phase
(or the weaker version: “it is always beneficial to find errors in an earlier phase”)

[simula . research laboratory]

More over-simplifications

- “Adding manpower to a late software project makes it later" *The Mythical Man-Month* (Brooks’s law)
- The difference in productivity between the best and the worst programmer is 10:1
- Avoid duplicating code (”*Number one in the stink parade*” - Martin Fowler, agile guru)

[simula . research laboratory]

Tools of a “myth buster” (part of evidence-based practice)

1. Find out what is meant by the claim.
   - Is it possible to falsify the claim? If not, what is the function of the claim?
2. Put yourself in a ”critical mode”
   - Raise the awareness of the tendency to accept claims, even without valid evidence, when you agree/it seems intuitively correct.
   - Consider what you would consider as valid evidence to support the claim.
   - Vested interests?
   - Do you agree because of the source?
3. Collect and evaluate evidence
   - Research-based, practice-base, and “own” evidence
4. Synthesize evidence and conclude (if possible)

HOW RATIONAL ARE OUR UNAIDED (NON EVIDENCE-BASED) JUDGEMENTS?

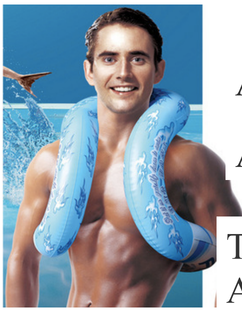
[simula . research laboratory]

Making a decision or choice makes the world look different

http://www.ted.com/talks/lang/eng/dan_gilbert_asks_why_are_we_happy.html

[simula . research laboratory]

Are risk-willing developers better?



Group A:

Average 1: 3.3

Average 2: 3.5

Two weeks later
Average 3: 3.5

Group B:

Average 1: 5.4

Average 2: 5.0

Two weeks later
Average 3: 4.9

Study design: Research evidence + Self-generated argument.

Question: Based on your experience, do you think that risk-willing programmers are better than risk-averse programmers?
1 (totally agree) – 5 (No difference) - 10 (totally disagree)

[simula . research laboratory]

What does convince software engineers?

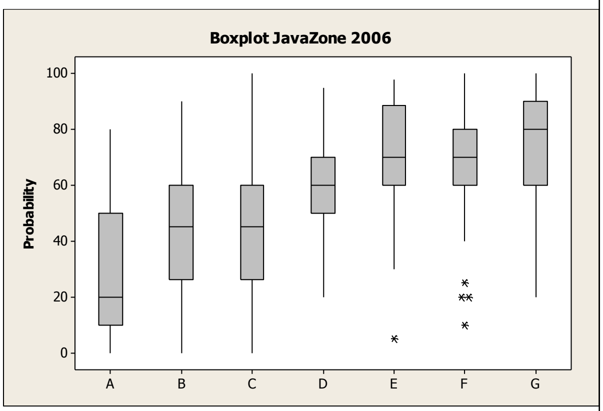
Context: Assume that a test course provider claims: "*The course will lead to substantial increase in test efficiency and quality for most participants.*"

How likely do you think this claim is true, given *[reduced explanation]*:
A: No other information
B: Supporting claims from reference clients
C: Supporting study conducted by the course provider
D: Convincing explanation (but no empirical evidence)
E: Supporting experience from a colleague (It helped him)
F: Supporting scientific study completed at a renowned university
G: Own experience (It helped me)

[simula . research laboratory]

The results

A: No other information
B: Support from reference clients
C: Supporting study conducted by the course provider
D: Convincing explanation (but no empirical evidence)
E: Supporting experience from a colleague (It helped him)
F: Supporting scientific study completed at a renowned university
G: Own experience (It helped me)



[simula . research laboratory]

"I see it when I believe it" vs "I believe it when I see it"

- **Design:**
  – Data sets with randomly set performance data comparing "traditional" and "agile" methods.
  – Survey of each developer's belief in agile methods
- **Question**: How much do you, based on the data set, agree in: *"Use of agile methods has caused a better performance when looking at the combination of productivity and user satisfaction."*
- **Result**:
  – Previous belief in agile determined what they saw in the random data

[simula . research laboratory]

Why do we believe in myths and oversimplifications?

- Meeting a need or desire. We want something to be true and don't look for disconfirming evidence.
- Lack of precision in claim or misunderstood research.
- We are more concerned about whether something sounds correct, than about asking what this really means.
- Self-fulfilling claims (we see it because we believe it).
- It is easier to find confirming and than to find representative evidence.
- Political and business-related reasons. Deliberate creation of myths.
  – Repetition.
  – Presented by authorities.
- To understand is to accept. De-accepting is more difficult

[simula . research laboratory]

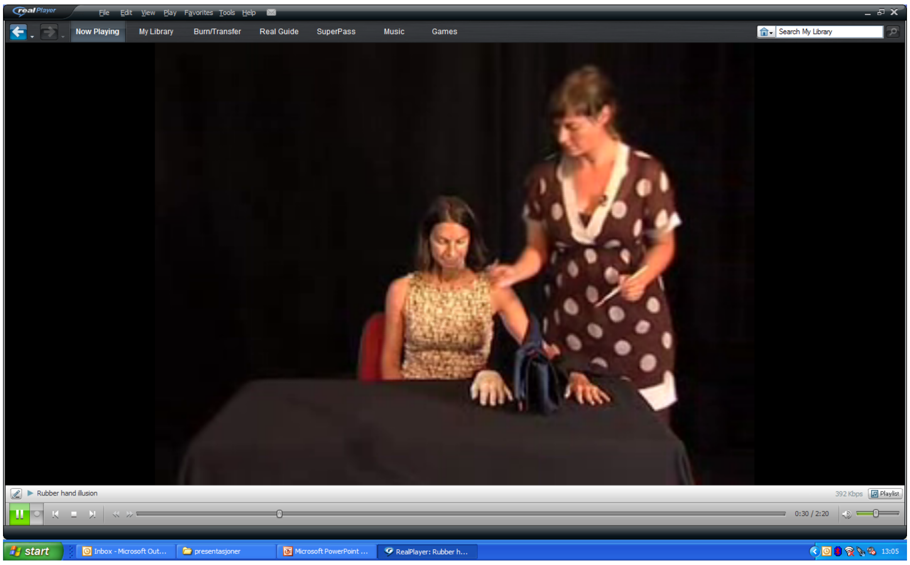
Final words ....

- When it is important to make the right decisions, base this on relevant and valid evidence.
- This evidence may be based on systematic collection of evidence from
  - research studies. **NB**: We are in the age of google scholar!
  - practice. Google, use your network, interest groups etc. to find people with representative and, if possible, neutral experience.
  - own studies. Collect your own evidence, e.g., from using more than one method, company, tool on the first increment of a project.
- Evaluate critically the evidence and synthesize it
- Train your ability to collect and evaluate evidence and argumentations
- Know about your own and other peoples typical interpretation and learning biases (first impression, confirmation bias, ...)

Extra:

What we should be aware of when collecting and evaluating practice-based experience

Learning problem 1: We see what we expect to see



[simula . research laboratory]

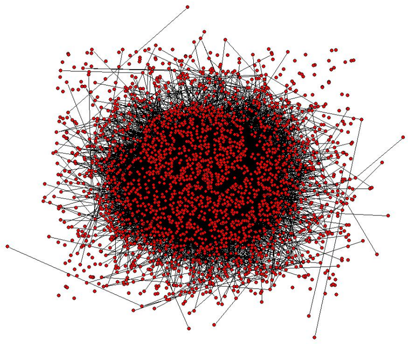
Learning problem 2: “We won” - “they lost”

- We sincerely believe that we succeeded because we are skilled and failed because we had bad luck.
- The need for a high level of self-esteem makes learning sometimes difficult.
- Example:
  - Software developers systematically point at reasons outside their control to explain failures, and reasons the control as reasons for success.

[simula . research laboratory]

Learning problem 3: Lack of the total picture

- **Local interpretation:** In a company, most project leaders agreed on that the most important reason for overruns was lack of clear and precise requirements.
- An analysis of the projects suggested the opposite. The advantage of vague requirements (increase of flexibility) was larger than the disadvantage of the lack of claritv.
- Exercise: Why didn't the project leaders discover this?

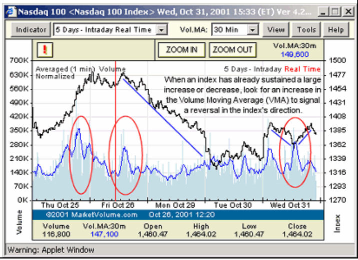


Learning problem 4: Superficial Learning

- Most people stop when they have believed they have found the direct causes, and do not look for indirect and contributory reasons.
  - A reason for problem failure is, for example, frequently ”unexpected events”.
  - BUT, unexpected events are quite common and should not be unexpected.
  - The important cause may be why they weren't sufficiently prepared for unexpected events.
- Children's way of asking are in many ways good learning examples for deeper learning.

Learning problem 5: We see patterns were there are none

- HOT HAND?
  - ”*Basketball players and fans alike tend to believe that a player’s chance of hitting a shot are greater following a hit than following a miss on the previous shot. However, detailed analyses of the shooting records of [reference to several studies and a controlled shooting experiment] provided no evidence for a positive correlation between the outcomes of successive shots.*” (Gilovich, COGNITIVE PSYCHOLOGY 17, 295-314, 1985)
- Frequently the same problem in IT-projects. If B follows A two times in a row, we have a rule.
- Stock market analysis is heavily based on finding patters where there are none.



Learning problem 6: Hindsight bias

- In a survey we gave the software professionals real and invented project outcomes. Regardless of the version they received, most of them thought that the outcomes were as expected.
- We do this, even when we (at least on behalf of others) are aware of the hindsight bias effect

Learning problem 7: Falsification

- Several studies show that we tend to confirm what we believe and are very poor at looking for and emphasizing non-conforming evidence.
- The consequence is that we may end up believing strongly in incorrect or strongly uncertain knowledge.
- Recent studies suggest that those looking for disconfirming evidence are the better testers!

Learning problem 8: A strong focus on "we should learn from this " may make it worse

- In particular, when the desire is not connected with the opportunities to learn
  - F. I. Steele: Organizational overlearning, Journal of Management Studies, 1971.
- Example: Governmental reports on the reasons for failed, mega-large IT-projects.
  - Interpretations based on highly incomplete argumentation
  - The causal chain is clearly too simplistic. There are, for example, many cases where the same chain led to success.
- Paradox: The learning itself frequently makes the learning less relevant.

[simula . research laboratory]

How should we collect reliable knowledge?

Guidelines: Check relevance, combine perspectives, triangulate of methods, be critical, design processes that go for the deeper cause-effect relationships

- Check the relevance of the experience. Remember that:
  1. Relevance of knowledge and skill can be very narrow.
  2. Experience is not the same as knowledge. Preferably, to transfer from experience to relevant and reliable knowledge, the following conditions should be met by the people's learning situation:
     - Learning-friendly conditions. Preferable situations where only few changes takes place and there are systematic effect measurement in place.
     - Unbiased interpretations. A person responsible for selecting a new tool is, as an illustration, not the best one to assess it's impact on quality and productivity.

How should we collect reliable knowledge?

- If unbiased, complete pictures from one person is difficult, try to collect information from more than one perspective, background and role.
  - Preferably, the informants should have formed their knowledge independent of each other.
- Example of knowledge collection technique:
  - Observations of on-the-job work
  - Interviews
  - Observations in controlled contexts with verbal protocols (thinking-aloud)
  - Study of written material (emails, experience reports, etc.)
  - Statistical modeling
  - Concept mapping
  - Sessions of analysis of cause-effects (Root Cause Analysis, Ishikawa, Post Mortem Analyses, …)

Final words ....

- When it is important to make the right decisions, we should base this on relevant and valid evidence.
- This evidence may be based on
  - research studies. We are in the age of google scholar!
  - practice. Google, use your network, interest groups etc. to find people with representative and, if possible, neutral experience.
  - own studies. Collect your own evidence, e.g., from using more than one method, company, tool on the first increment of a project.
- Evaluate critically the evidence and synthesize it
- Train your ability to collect and evaluate evidence and argumentations
- Know about your own and other peoples typical interpretation and learning biases (first impression, confirmation bias, ...)

[simula . research laboratory]