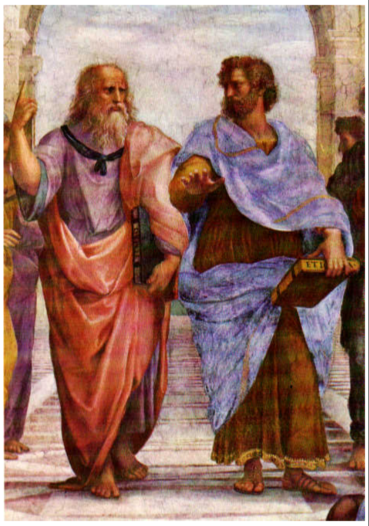
[simula . research laboratory]

Ten years with Evidence-Based Software Engineering

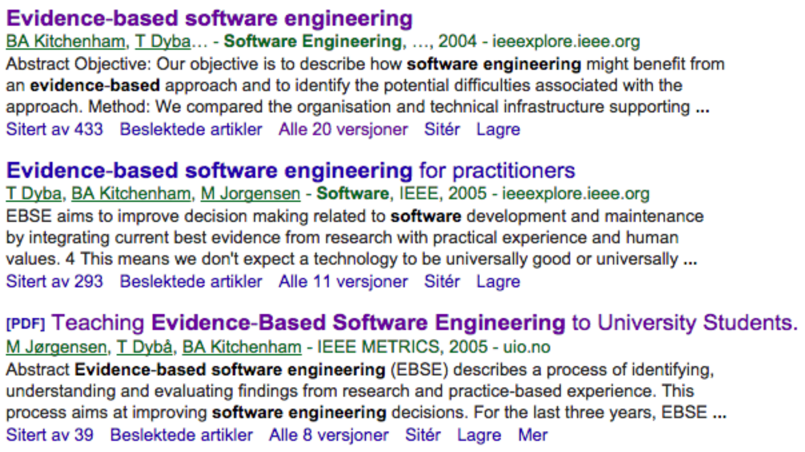
What is it?
Has it had any impact?
What's next?

Magne Jørgensen



Initial papers on evidence-based software engineering

Barbara Kitchenham
Tore Dybå
Magne Jørgensen



Evidence-based software engineering
BA Kitchenham, T Dyba… - **Software Engineering**, …, 2004 - ieeexplore.ieee.org
Abstract Objective: Our objective is to describe how **software engineering** might benefit from an **evidence-based** approach and to identify the potential difficulties associated with the approach. Method: We compared the organisation and technical infrastructure supporting **...**
Sitert av 433 Beslektede artikler Alle 20 versjoner Sitér Lagre

Evidence-based software engineering for practitioners
T Dyba, BA Kitchenham, M Jorgensen - **Software**, IEEE, 2005 - ieeexplore.ieee.org
EBSE aims to improve decision making related to **software** development and maintenance by integrating current best evidence from research with practical experience and human values. 4 This means we don't expect a technology to be universally good or universally **...**
Sitert av 293 Beslektede artikler Alle 11 versjoner Sitér Lagre

[PDF] Teaching **Evidence-Based Software Engineering** to University Students.
M Jørgensen, T Dybå, BA Kitchenham - IEEE METRICS, 2005 - uio.no
Abstract **Evidence-based software engineering** (EBSE) describes a process of identifying, understanding and evaluating findings from research and practice-based experience. This process aims at improving **software engineering** decisions. For the last three years, EBSE **...**
Sitert av 39 Beslektede artikler Alle 8 versjoner Sitér Lagre Mer

[simula . research laboratory]

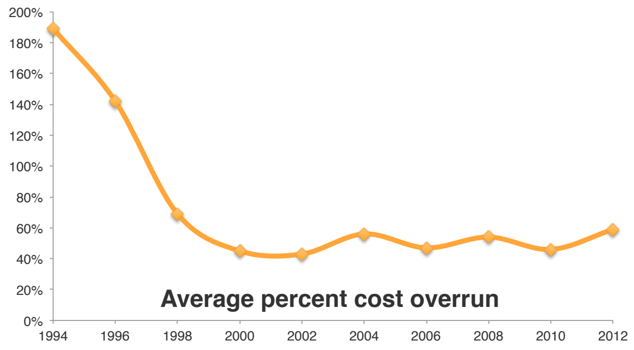
Evidence-based software engineering (EBSE)

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. (+ create own local evidence, if needed)
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.

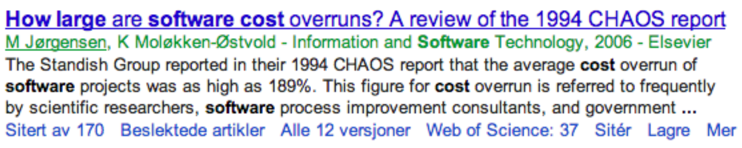
Most (93%) of our communication is non-verbal

Software Chaos? Huge improvement 1994-1998?



(page 13 of their 1994 CHAOS 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 (but possible) to debunk myths

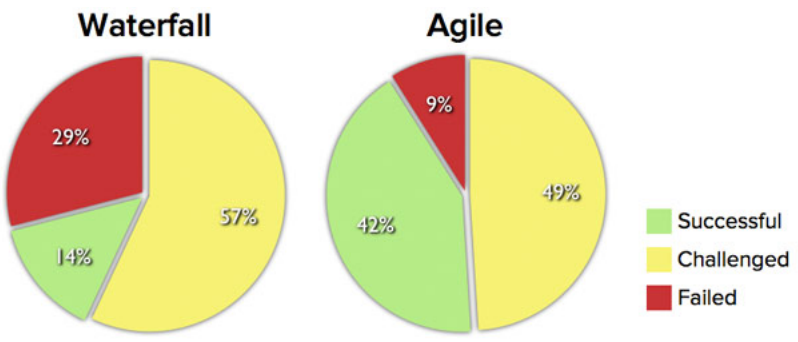


How large are **software cost** overruns? A review of the 1994 CHAOS report
M Jørgensen, K Moløkken-Østvold - Information and **Software** Technology, 2006 - Elsevier
The Standish Group reported in their 1994 CHAOS report that the average **cost** overrun of **software** projects was as high as 189%. This figure for **cost** overrun is referred to frequently by scientific researchers, **software** process improvement consultants, and government **...**
Sitert av 170 Beslektede artikler Alle 12 versjoner Web of Science: 37 Sitér Lagre Mer

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.

But new ones are arriving all the time …

14% Waterfall and 42% of Agile projects are successful
(source: The Standish Group, The Chaos Manifesto 2012)



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

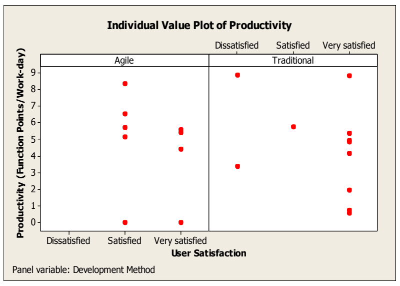
Successful = "On cost, on schedule and with specified functionality"

Can you spot a serious error in this comparison?

[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]

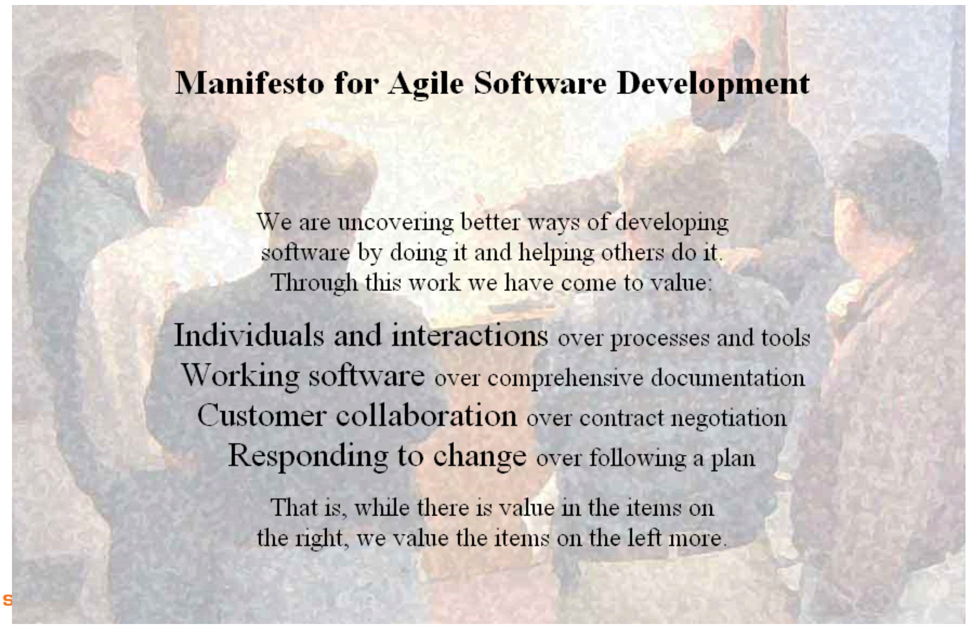
WHAT ARE THE SKILLS NEEDED TO PRACTICE EVIDENCE-BASED SOFTWARE ENGINEERING?

Formulate questions meaningful for your context/challenge/problem

The question “Is Agile better than Traditional methods?” is NOT answerable.

- What is agile?
- What is traditional?
- What is better?
- What is the context?

Question what statements and claims means



Manifesto for Agile Software Development

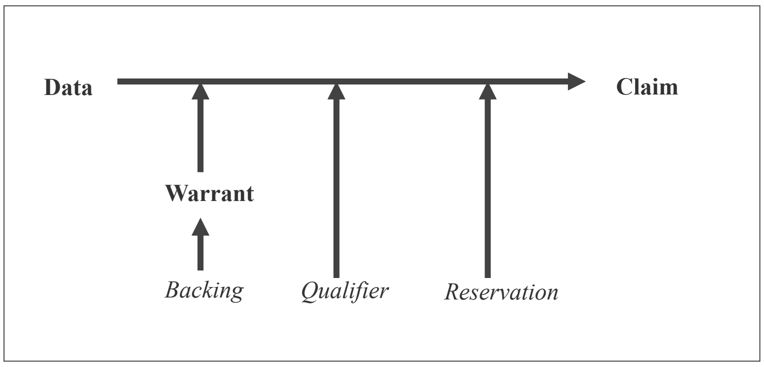
We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

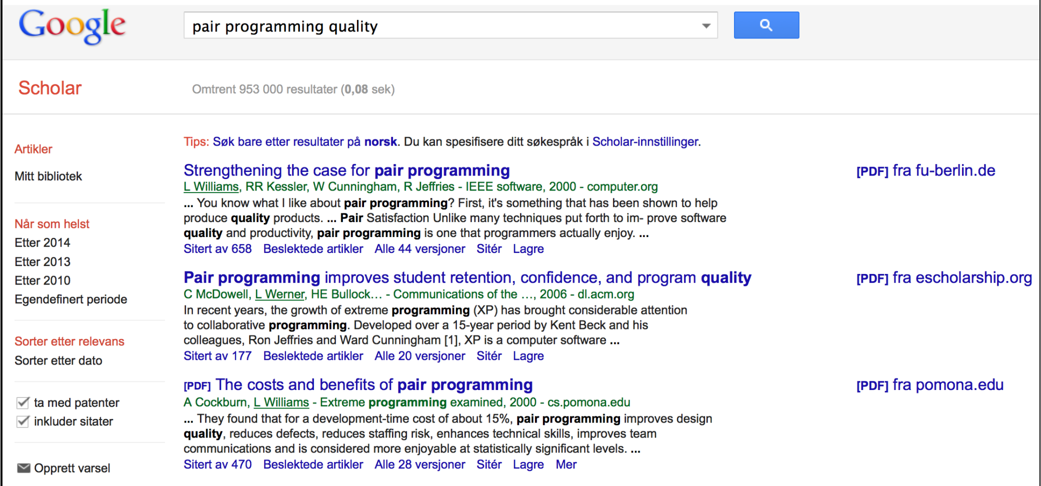
[s

Know how to evaluate argumentation



[simula . research laboratory]

Know how to use google scholar (search for research-based evidence)



Omtrent 953 000 resultater (**0,08** sek)

Artikler
Mitt bibliotek

Når som helst
Etter 2014
Etter 2013
Etter 2010
Egendefinert periode

Sorter etter relevans
Sorter etter dato

☑ ta med patenter
☑ inkluder sitater

Opprett varsel

Tips: Søk bare etter resultater på **norsk**. Du kan spesifisere ditt søkespråk i Scholar-innstillinger.

Strengthening the case for **pair programming** — **[PDF]** fra fu-berlin.de
L Williams, RR Kessler, W Cunningham, R Jeffries - IEEE software, 2000 - computer.org
... You know what I like about **pair programming**? First, it's something that has been shown to help produce **quality** products. ... **Pair** Satisfaction Unlike many techniques put forth to im- prove software **quality** and productivity, **pair programming** is one that programmers actually enjoy. ...
Sitert av 658 Beslektede artikler Alle 44 versjoner Sitér Lagre

Pair programming improves student retention, confidence, and program **quality** — **[PDF]** fra escholarship.org
C McDowell, L Werner, HE Bullock... - Communications of the ..., 2006 - dl.acm.org
In recent years, the growth of extreme **programming** (XP) has brought considerable attention to collaborative **programming**. Developed over a 15-year period by Kent Beck and his colleagues, Ron Jeffries and Ward Cunningham [1], XP is a computer software ...
Sitert av 177 Beslektede artikler Alle 20 versjoner Sitér Lagre

[PDF] The costs and benefits of **pair programming** — **[PDF]** fra pomona.edu
A Cockburn, L Williams - Extreme **programming** examined, 2000 - cs.pomona.edu
... They found that for a development-time cost of about 15%, **pair programming** improves design **quality**, reduces defects, reduces staffing risk, enhances technical skills, improves team communications and is considered more enjoyable at statistically significant levels. ...
Sitert av 470 Beslektede artikler Alle 28 versjoner Sitér Lagre Mer

[simula . research laboratory]

Know how to collect and evaluate practice-based experience

- Methods similar to evaluation of research-based evidence and claims
- Be aware of "organizational over-learning"

[simula . research laboratory]

Know how to create local evidence

- Experimentation is often possible and creates the best evidence
  - Pilot studies
  - Trial-sourcing
  - Controlled experiments in own context

Any impact from EBSE?

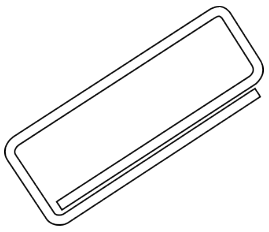
- Successes:
  - Trained several hundreds of software engineers (both students and professionals)
  - Courses in evidence-based software engineering (or closely related courses) created at several universities
  - Conferences and research projects with this as core topic
  - Upcoming journal (within information systems) with this as topic
- Failures:
  - No systematic evaluation of how well software engineers are able to implement and how often they actually use the EBSE-steps
  - No real success story to tell (Company X used this and it led to ……)

What's next

- **Still to be answered**: Is it realistic to achieve an evidence-based software engineering profession, or is this only for those with research competence?
- **My opinion**: Moderate optimist based on my experience with teaching this to students and professionals.
- Challenges:
  - Not much research (evidence-based medicine is easier)
  - High number of different contexts makes use of evidence complex
  - Systematic literature reviews are typically too much "we need more research" + student experiments and not very useful for the software industry
- Opportunities:
  - More emphasis on the teaching how to collect practice-based evidence (and less on systematic reviews, google scholar-searches)
  - More emphasis on the generation of local evidence (teaching of how to do controlled trials in own context)

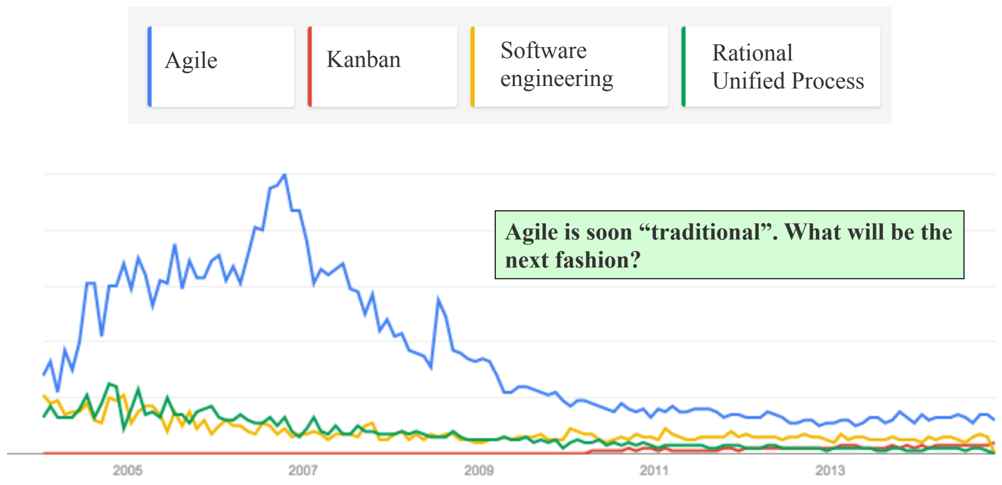
[simula . research laboratory]

The paper clip was invented by a Norwegian



[simula . research laboratory]

Google trends (fashions): Worldwide web searches **Category**: Computers & Electronics, Software



Agile is soon "traditional". What will be the next fashion?

[simula . research laboratory]

The ease of creating myths: Are risk-willing or risk-averse developers better?

Group A:

Initially
Average 3.3

Debriefing
Average 2: 3.5

2 weeks later
Average 3: 3.5

Group B:

Initially
Average 5.4

Debriefing
Average 2: 5.0

2 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)

Neutral group: Average 5.0

[simula . research laboratory]