INF 5500

Empirical Methods and Evidence-based Decisions in Software Engineering

Magne Jørgensen

magnej@simula.no

Introduction

Learning goals of this lecture:

- Understand the goals of the course and how to get a good grade.
- Knowledge about the main steps of systematic, evidence-based decision processes.
- Introduction to the importance of critical appraisal of argumentation.

Supporting text:

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

Course Assumptions and Goals

- **ASSUMPTION**: Important decisions in software engineering should, as far as possible, be based on critical collection and evaluation of research results and practice-based experience.
- Critical reflections:
  - Is this a reasonable assumption?
  - To what degree are there evidence to support the use of evidence-based and science-based principles?
- **LEARNING GOAL**: Increased ability to base important [software engineering] decisions on collection and critical appraisal of available research and practice-based evidence.
- Critical reflections:
  - Is it reasonable to believe that you will be able to transfer an increased ability in a class room setting/learning from a project report to the “real world”?

Lecture Content (1)

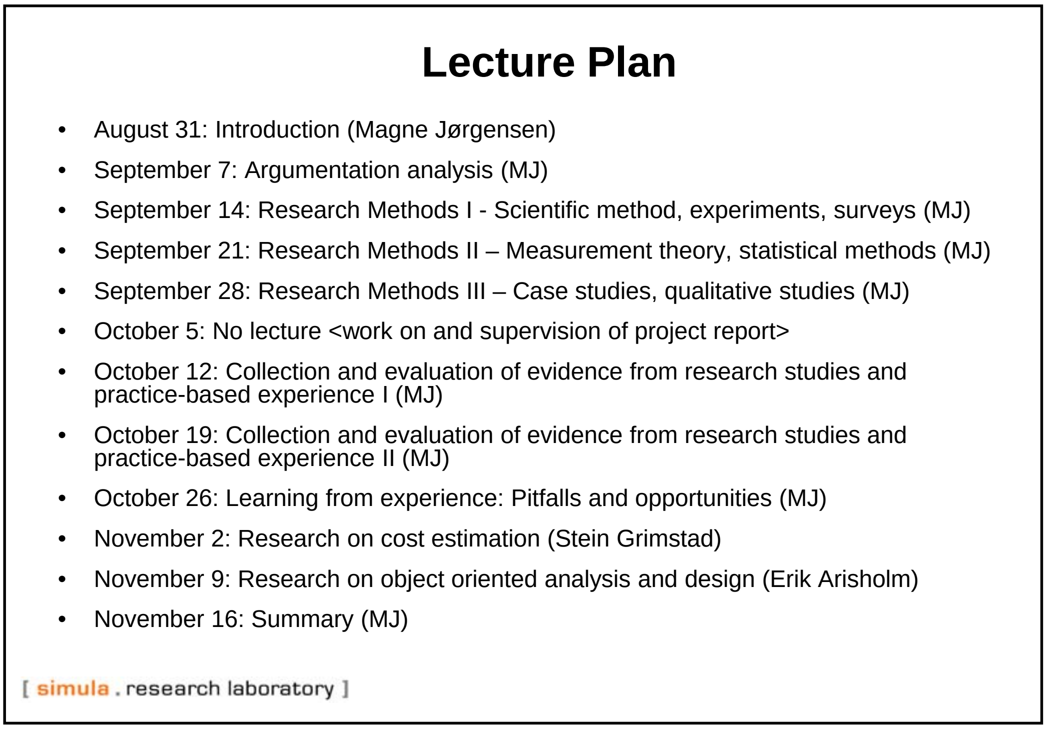
- Lecture 1: Evidence-based Software Engineering
  - Tore Dybå, Barbara Kitchenham and Magne Jørgensen, Evidence-based Software Engineering for Practitioners, IEEE Software, Vol. 22, No. 1, Jan-Feb 2005.
- Lecture 2: Argumentation analysis
  - Alec Fisher, The logic of real arguments, Chapter 2: A general method of argument analysis. Cambridge University Press. 2004. p 15-28.
  - Karyn Charles Rybacki and Donald Jay Rybacki, Advocacy and opposition, Chapter 8: What should I avoid? Pearson. 2004. p 142-163.
  - Briony J Oates, Researching information systems and computing, SAGE Publications. (Section 7, Surveys)
  - Claes Wohlin et al. Experimentation in software engineering. Kluwer Academic Publishers. (Section 11, Experiments)
  - www.unc.edu/depts/wcweb/handouts/evidence_use.html

Lecture Content (2)

- Lectures 3-5: Empirical research methods
  - General introduction to scientific method: www.freeinquiry.com/intro-to-sci.html
  - Briony J Oates, Researching information systems and computing, SAGE Publications.
  - Claes Wohlin et al. Experimentation in software engineering. Kluwer Academic Publishers.
  - Empirical research in software engineering: Barbara Kitchenham et al., Preliminary Guidelines for Empirical Research in Software Engineering, IEEE Transactions on Software Engineering, 2002.
  - Statistical studies of treatment-effects: www.moffitt.org/moffittapps/ccj/v4n5/article4.html

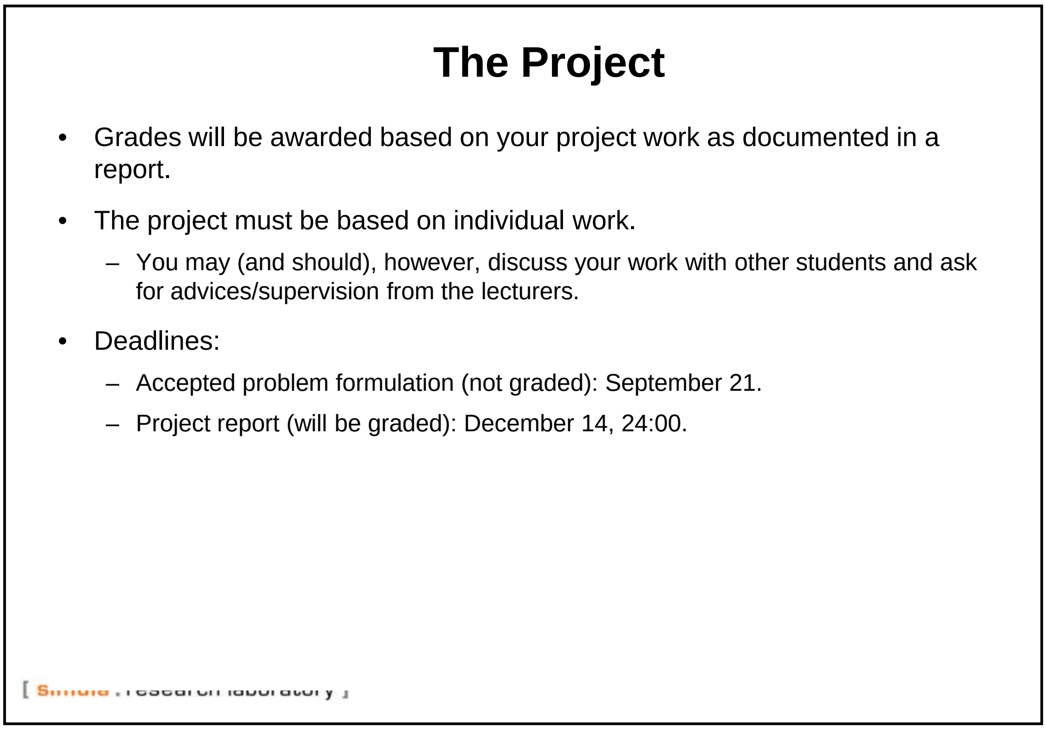
Lecture Content (3)

- Lectures 6-8: Collection and evaluation of results from research studies and experience-based practice
  - www.ub.uio.no/umn/inf/
  - www.skepdic.com/essays/evaluatingexperience.html
  - http://changingminds.org/index.htm
  - Shari Lawrence Pfleeger, Soup or Art? The Role of Evidential Force in Empirical Software Engineering.
- Lectures 9-10: Examples of research designs, research results and collection and evaluation of results from research studies.
- Lecture 11: Summary, questions.



Lecture Plan

- August 31: Introduction (Magne Jørgensen)
- September 7: Argumentation analysis (MJ)
- September 14: Research Methods I - Scientific method, experiments, surveys (MJ)
- September 21: Research Methods II – Measurement theory, statistical methods (MJ)
- September 28: Research Methods III – Case studies, qualitative studies (MJ)
- October 5: No lecture <work on and supervision of project report>
- October 12: Collection and evaluation of evidence from research studies and practice-based experience I (MJ)
- October 19: Collection and evaluation of evidence from research studies and practice-based experience II (MJ)
- October 26: Learning from experience: Pitfalls and opportunities (MJ)
- November 2: Research on cost estimation (Stein Grimstad)
- November 9: Research on object oriented analysis and design (Erik Arisholm)
- November 16: Summary (MJ)



The Project

- Grades will be awarded based on your project work as documented in a report.
- The project must be based on individual work.
  - You may (and should), however, discuss your work with other students and ask for advices/supervision from the lecturers.
- Deadlines:
  - Accepted problem formulation (not graded): September 21.
  - Project report (will be graded): December 14, 24:00.

The Project Report Should Include (1)

1) A problem formulation relevant for software development. The problem formulation may, for example, be related to a choice situation or a claim of interest for software professionals.

- Example of a choice situation: An organization consider replacing C++ with Java. A problem formulation should include the purpose of the change, relevant properties of the organization, the meaning of essential terms (e.g., improved efficiency), etc.
- **A good problem formulation is essential for the subsequent work.**

2) A description of your **systematic** search for research and practice related experience relevant for the problem formulation and the identified information (with complete reference to the sources).

- The amount of relevant research for some problem formulation may be limited to quite few studies. In that case, the collection of larger amount of practice related experience will be essential, e.g., by contacting software professionals with relevant experience.
- It is required to apply research literature data bases (i.e., information based on “googling” only will lead to the grade F).

The Project Report Should Include (2)

3) A description of your critical evaluation of relevance and validity of each of the included information sources.

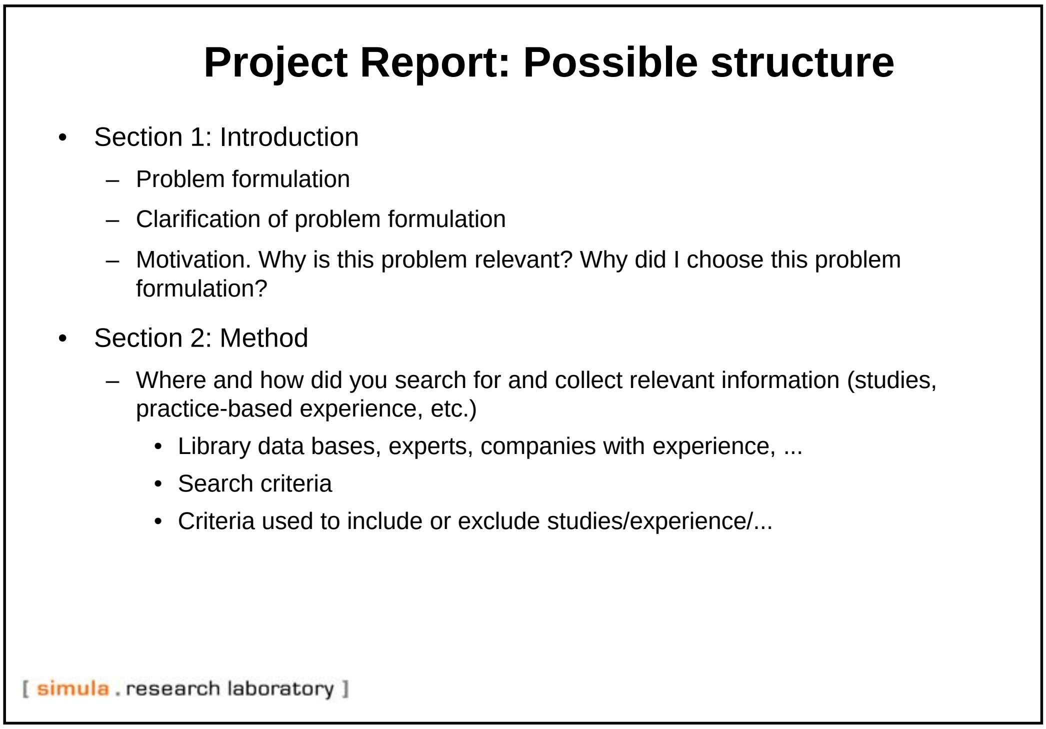
- Clearly biased and low quality information sources need no thorough evaluation.

4) A synthesis of the information as basis for an evidence-based conclusion related to the problem formulation.

5) An outline of a research study, including design rationale, that builds upon existing evidence and addresses important aspects of the problem you formulated.

The report should be maximum 20 pages long and be written in English or Norwegian.

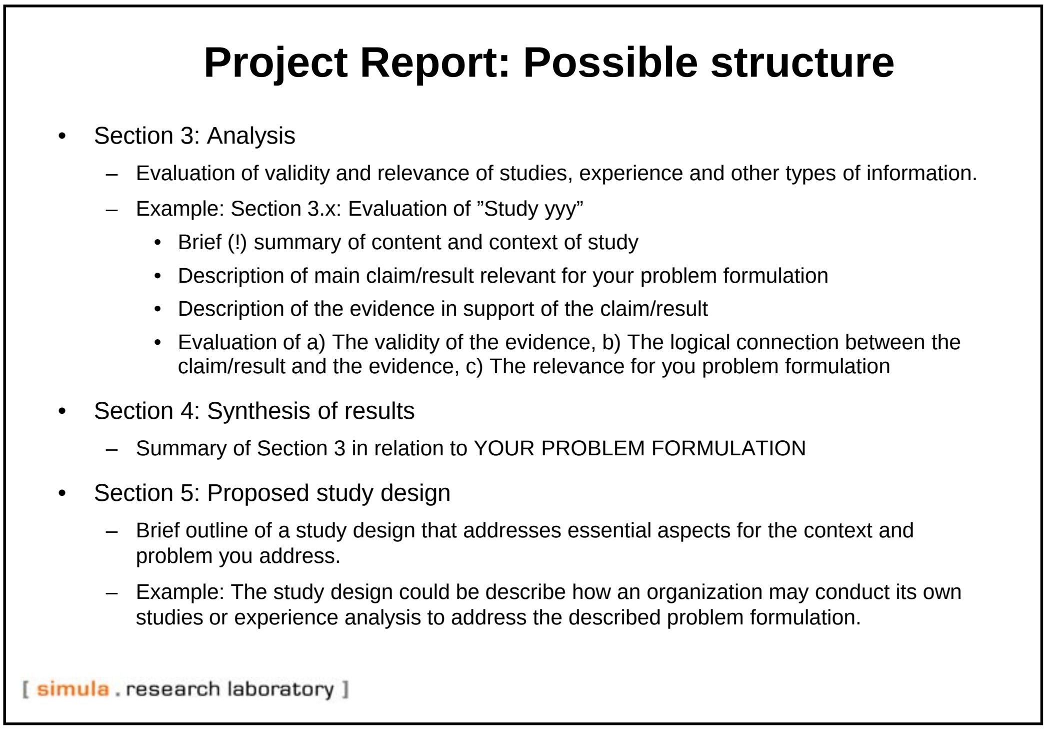
NB: Those of you following this as a PhD course, need to provide a 20% more extensive project report. The quality criteria and requirements will be the same.



Project Report: Possible structure

- Section 1: Introduction
  - Problem formulation
  - Clarification of problem formulation
  - Motivation. Why is this problem relevant? Why did I choose this problem formulation?
- Section 2: Method
  - Where and how did you search for and collect relevant information (studies, practice-based experience, etc.)
    - Library data bases, experts, companies with experience, ...
    - Search criteria
    - Criteria used to include or exclude studies/experience/...

[simula . research laboratory]



Project Report: Possible structure

- Section 3: Analysis
  - Evaluation of validity and relevance of studies, experience and other types of information.
  - Example: Section 3.x: Evaluation of ”Study yyy”
    - Brief (!) summary of content and context of study
    - Description of main claim/result relevant for your problem formulation
    - Description of the evidence in support of the claim/result
    - Evaluation of a) The validity of the evidence, b) The logical connection between the claim/result and the evidence, c) The relevance for you problem formulation
- Section 4: Synthesis of results
  - Summary of Section 3 in relation to YOUR PROBLEM FORMULATION
- Section 5: Proposed study design
  - Brief outline of a study design that addresses essential aspects for the context and problem you address.
  - Example: The study design could be describe how an organization may conduct its own studies or experience analysis to address the described problem formulation.

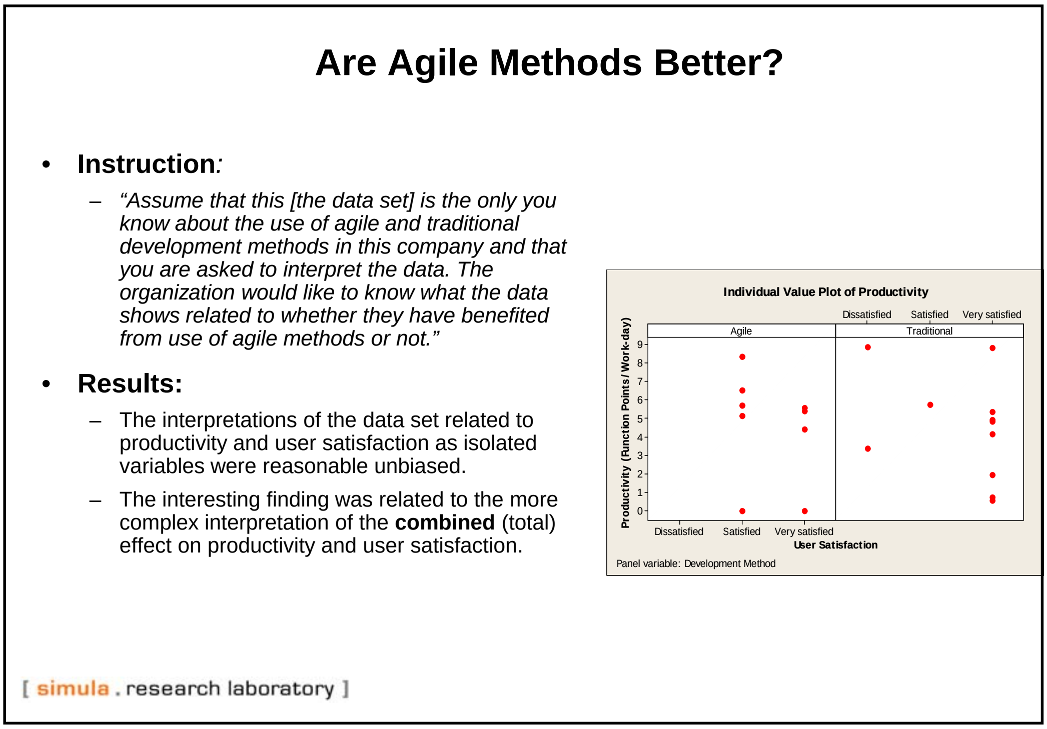
[simula . research laboratory]

Report Evaluation Criteria

- Quality of problem formulation (relevance, clarity, decidability, …)
- Quality and breath of information search (systematic, comprehensive, ….)
- Maturity of evaluation of the collected information
- Validity of synthesis of collected and evaluated information
- Quality and relevance of design of empirical study

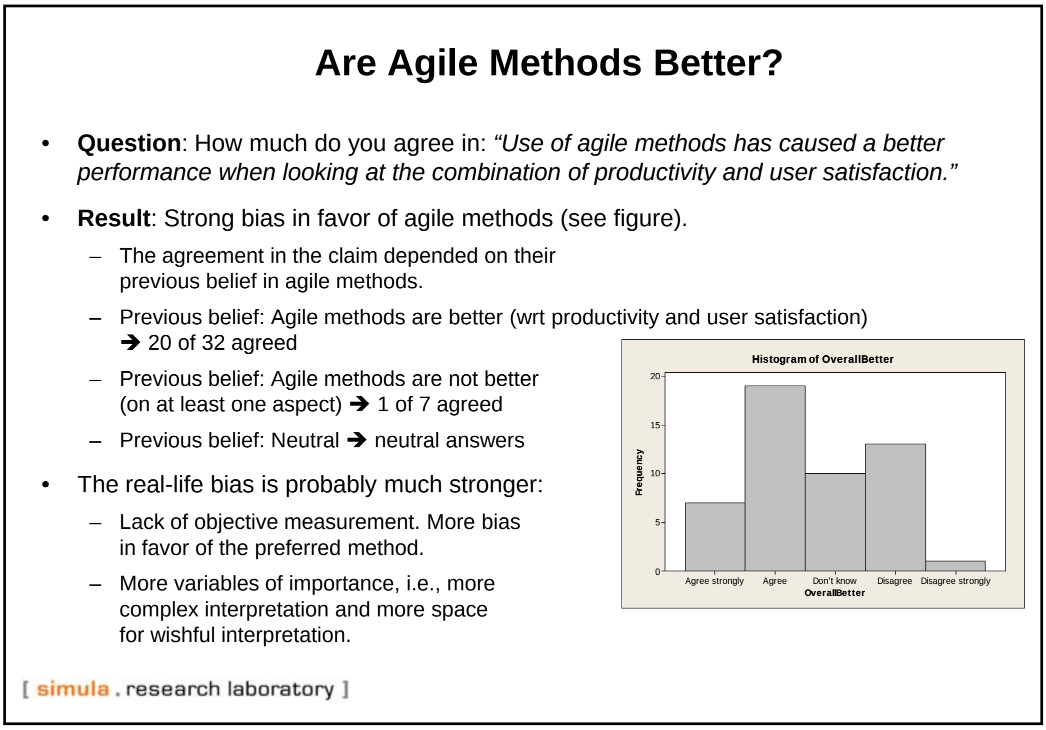
Why We Need Evidence-based Practices Are Agile Methods Better?

- **Participants**: 50 developers from a Polish company.
- **Strong belief in agile**: Before the study I collected their believes about agile methods.
  - 84% believed agile methods led to higher productivity (only 6% believed same or lower productivity), and 66% believed it led to more user satisfaction (only 8% same or lower).
- **Design of study**:
  - Generation of 10 project data sets (see example next page) with the triples: Development method (agile or traditional), Productivity (FP per work-day), and, User satisfaction (dissatisfied, satisfied, very satisfied).
  - All values were RANDOMLY generated.
  - A control gave that there were no (statistically) significant differences in the average values. The average values were slightly in favor of the traditional (non-agile) methods.
  - Each developer was randomly allocated to one of the data sets and asked to interpret it – **based on the measured data alone**.



Are Agile Methods Better?

- **Instruction**:
  - *“Assume that this [the data set] is the only you know about the use of agile and traditional development methods in this company and that you are asked to interpret the data. The organization would like to know what the data shows related to whether they have benefited from use of agile methods or not.”*
- **Results:**
  - The interpretations of the data set related to productivity and user satisfaction as isolated variables were reasonable unbiased.
  - The interesting finding was related to the more complex interpretation of the **combined** (total) effect on productivity and user satisfaction.



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.”*
- **Result**: Strong bias in favor of agile methods (see figure).
  - The agreement in the claim depended on their previous belief in agile methods.
  - Previous belief: Agile methods are better (wrt productivity and user satisfaction) ➔ 20 of 32 agreed
  - Previous belief: Agile methods are not better (on at least one aspect) ➔ 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.

Why Systematic Evaluations?

- The most common decision method in software development is based on “gut feeling” (intuition, expert judgment, unconscious mental processes). This method has many strengths:
  - We believe in the outcome (frequently essential for commitment)
  - It can be very fast and inexpensive (do not require data collection)
  - It is sometimes just as good as more scientific methods (no methods are free from subjectivity and biases)
- Pure judgment (not following a systematic, scientific process) has, however, limitations:
  - We have no access to the real argumentation. (We are, however, very good at rationalizing.)
  - People are sometimes strongly impacted by “wishful thinking” and other judgmental biases, WITHOUT knowing about it.
  - Judgmental processes are typically easy to manipulate (by sellers and gurus)
  - Important information may be missing due to lack of systematic search.
- When it is important to make the right decision, expert judgment should frequently not be the only decision method. We need systematic approaches based on scientific method.

Software professionals seem to rely very much on own and other people’s judgments

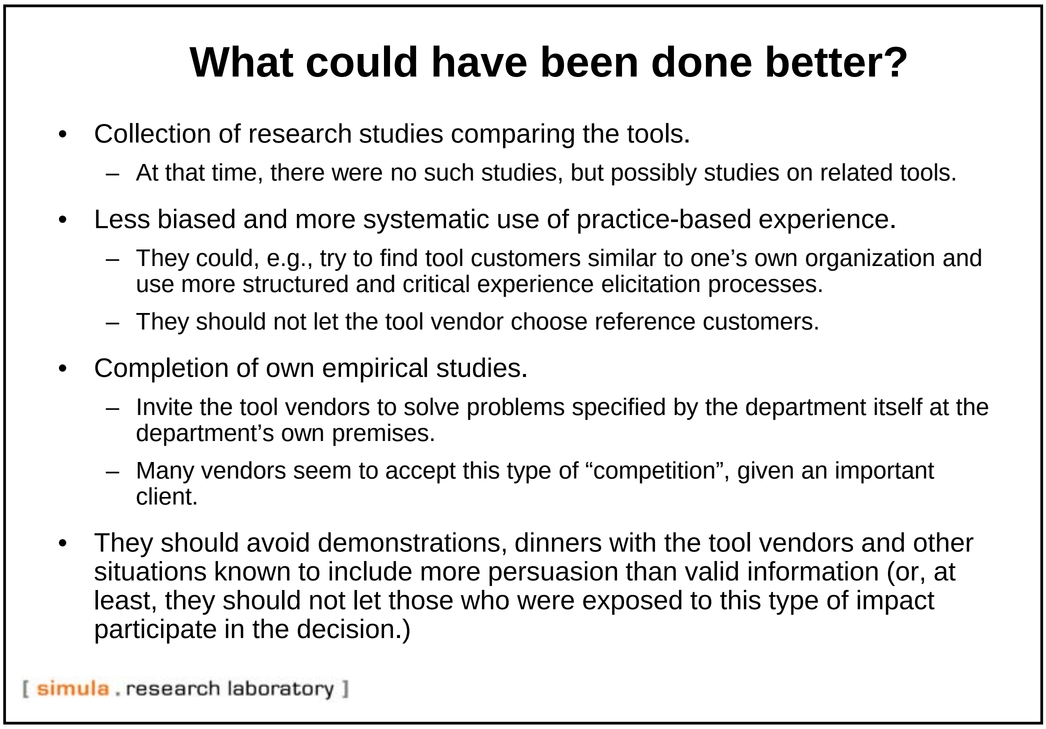
- Experiment (unpublished):
  - **Subjects**: 52 software professionals
  - **Context**: Evaluation of a course in software testing.
  - **Question**: How much do you agree in the statement: “*most of the participants of this testing course will substantially increase their efficiency and quality of test work*”.
  - **Treatment**: Different types of supportive evidence.
  - **Results**: As much as 15% reported that they would emphasize a positive course evaluation of a friend who had participated in the course more than supporting evidence from an independent study conducted by scientific researchers at a well-known university. If they themselves had participated and found the course useful, as many as 80% would believe more in their own, specific experience, than in the scientific study providing aggregated information.
  - **Implication**: This experiment illustrates that even in situations where the normative response would be to use the aggregated and more objective information, many people seem to prefer the highly specific.

What is valid evidence? A real-life example (1)

- A software development department wanted to replace their old-fashioned development tool with a more modern and more efficient one.
- They visited many possible vendors, participated at numerous demonstrations, and contacted several “reference customers”. Finally, they chose a development tool. The change cost about 10-20 million NOK + training and other indirect costs.
- A couple of years after the change, the department measured the change in development efficiency (not common – most software organizations never study the effect of their choices).
- Unfortunately, the development efficiency had not improved and the new development tool was far from as good as expected.
- This illustrated that even when applying much resources and time to collect evidence, software professionals may fail in making good decisions. What went wrong in this case?

What went wrong? A real-life example (2)

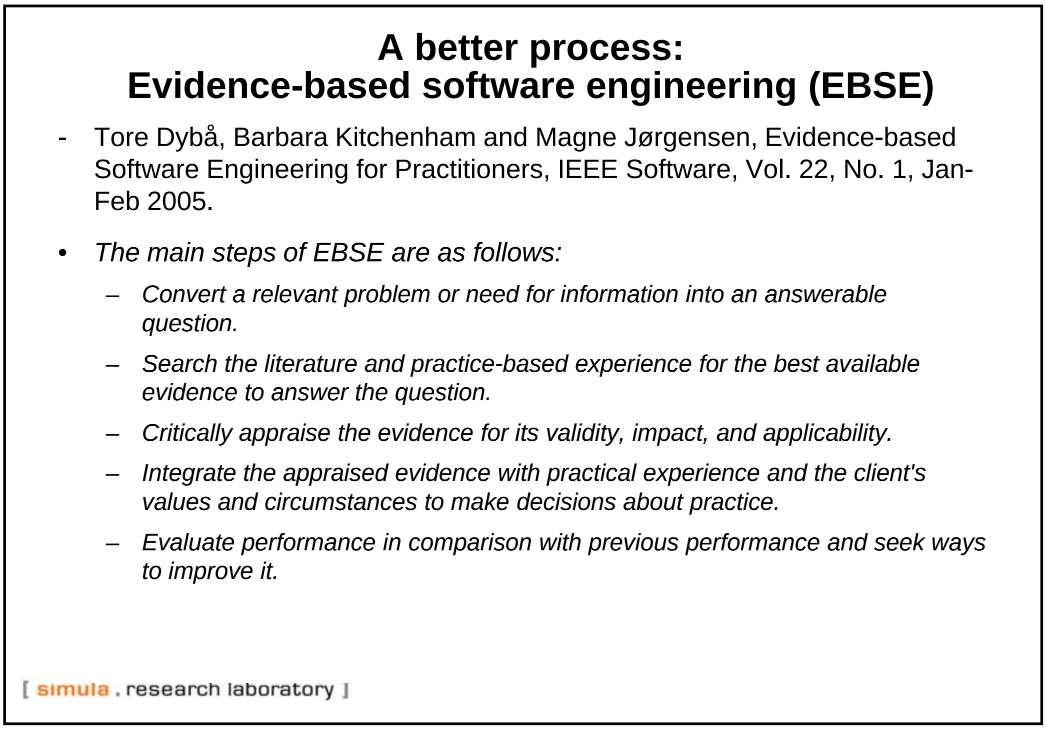
- The collection and evaluation of evidence had focused on “tool functionality”, following the principle “the more functionality, the better”.
- The demonstrations focused on strengths of the tools, not on weaknesses. Although, the software professionals were aware of this, they probably failed to compensate for what the demonstrations did not demonstrate. (We are not good at identifying lacking information!)
- The reference customers had themselves invested much money in the new tool. As long as they do not plan to replace the tool, then they would however not be reference customers anymore, they will tend to defend their decisions. (Avoidance of cognitive dissonance.)
- Although the amount of information (evidence) was high, they organization lacked the most essential information (independent evaluations of the tools in context similar to their own) and processes for critical evaluation of the information.
- In addition, they lacked the awareness of how they were impacted by the tool vendors persuasion techniques.
- Guidance in the principles of evidence-based software engineering would, we think, improved the decision.



What could have been done better?

- Collection of research studies comparing the tools.
  - At that time, there were no such studies, but possibly studies on related tools.
- Less biased and more systematic use of practice-based experience.
  - They could, e.g., try to find tool customers similar to one's own organization and use more structured and critical experience elicitation processes.
  - They should not let the tool vendor choose reference customers.
- Completion of own empirical studies.
  - Invite the tool vendors to solve problems specified by the department itself at the department's own premises.
  - Many vendors seem to accept this type of "competition", given an important client.
- They should avoid demonstrations, dinners with the tool vendors and other situations known to include more persuasion than valid information (or, at least, they should not let those who were exposed to this type of impact participate in the decision.)

[simula . research laboratory]



A better process: Evidence-based software engineering (EBSE)

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

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

[simula . research laboratory]

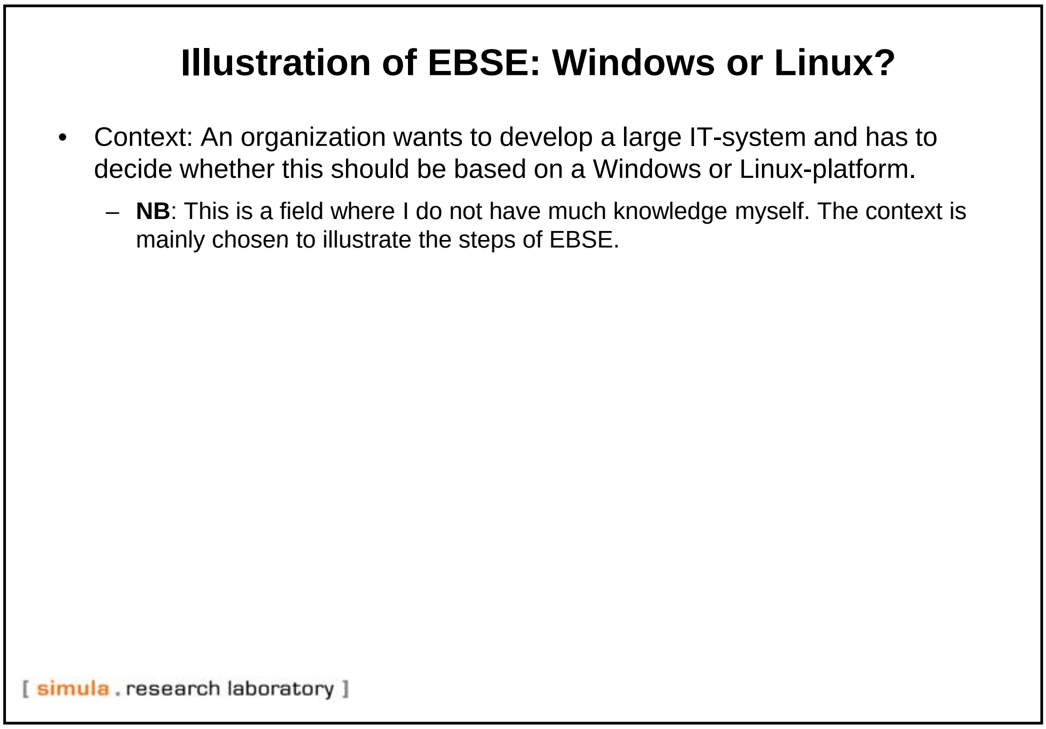
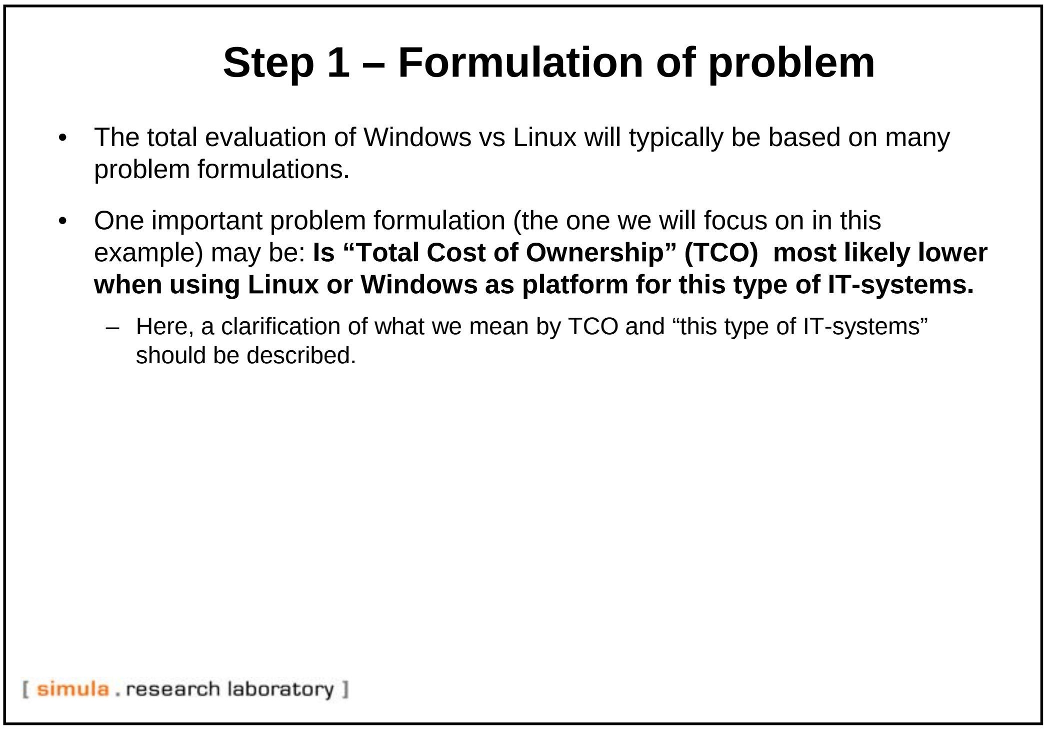


Illustration of EBSE: Windows or Linux?

- Context: An organization wants to develop a large IT-system and has to decide whether this should be based on a Windows or Linux-platform.
  - **NB**: This is a field where I do not have much knowledge myself. The context is mainly chosen to illustrate the steps of EBSE.



Step 1 – Formulation of problem

- The total evaluation of Windows vs Linux will typically be based on many problem formulations.
- One important problem formulation (the one we will focus on in this example) may be: **Is "Total Cost of Ownership" (TCO) most likely lower when using Linux or Windows as platform for this type of IT-systems.**
  - Here, a clarification of what we mean by TCO and "this type of IT-systems" should be described.

Step 2 – Collection of knowledge

Examples of search facilities:

- IEEE Xplore (http://ieeexplore.ieee.org) provides access to IEEE publications published since 1988.
- The IEEE Computer Society Digital Library (www.computer.org/publications/dlib) provides access to 22 IEEE Computer Society magazines and journals and more than 1,200 conference proceedings.
- The ACM Digital Library (www.acm.org/dl) provides access to ACM publications and related citations.
- The ISI Web of Science (www.isinet.com/products/citation/wos) consists of databases containing information from approximately 8,700 journals in different research areas.
- EBSCOhost Electronic Journals Service (http://ejournals.ebsco.com) provides access to over 8,000 e-journals.
- CiteSeer (http://citeseer.nj.nec.com), sponsored by the US National Science Foundation and Microsoft Research, indexes PostScript and PDF files of scientific research articles on the Web. Access is free.
- Google Scholar (http://scholar.google.com) indexes scholarly literature from all research areas, including abstracts, books, peer-reviewed papers, preprints, technical reports, and theses.

NB: If there are many information sources, focus on those published in journals of high quality and particularly reviews.

Step 2 – Collection of information

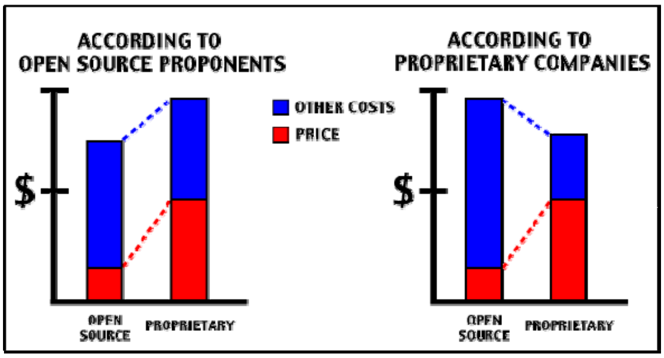
- If there are no/little documented experience/knowledge
  - Identify people, organizations and companies with relevant experience and ask them to provide information. This is in my experience easier than it at first sight may seem to be.
  - Emphasize representativeness, relevance and people without too much vested interests.
- DO NOT base the information collection on
  - random searches on the web and reading of the 4-5 first hits
  - reference clients chosen by the vendors
  - studies where there are strong vested interests

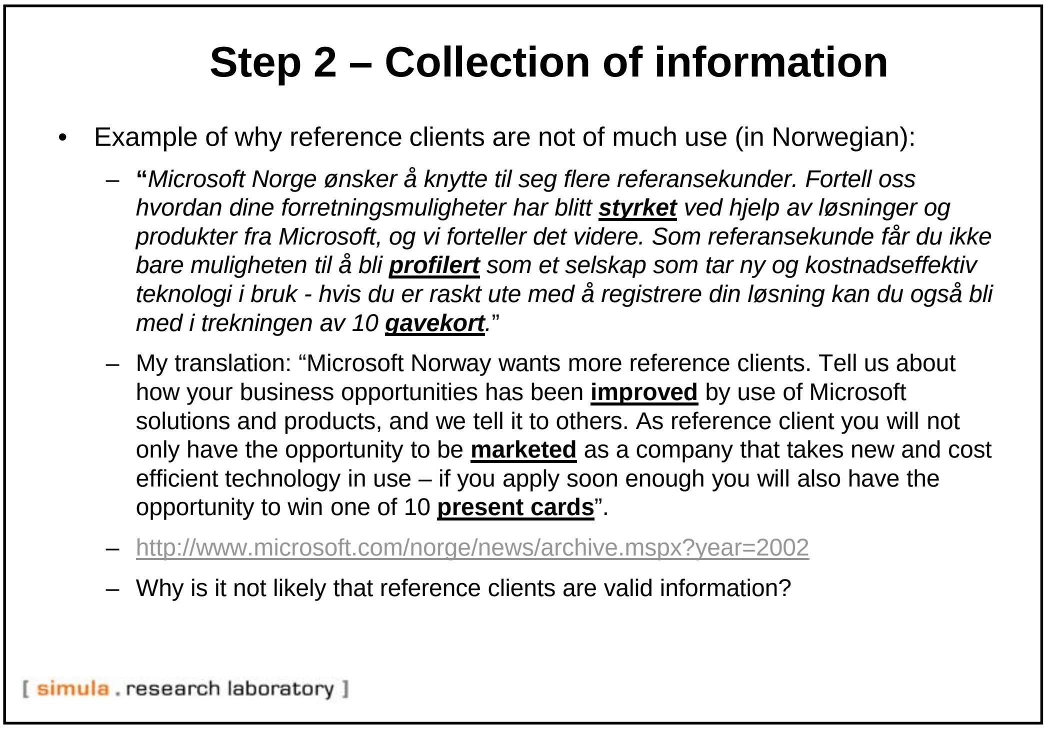
Step 2 – Collection of information

- My search using “Google scholar”:
  - *Windows AND Linux AND “total cost of ownership” [AND review].*
- Many hits. My strategy to filter the hits was in this case:
  - All analyses completed by organizations with strong vested interests were excluded.
  - Only analyses were it was likely that the author had competence in empirical studies were included.

Step 2 – Collection of information

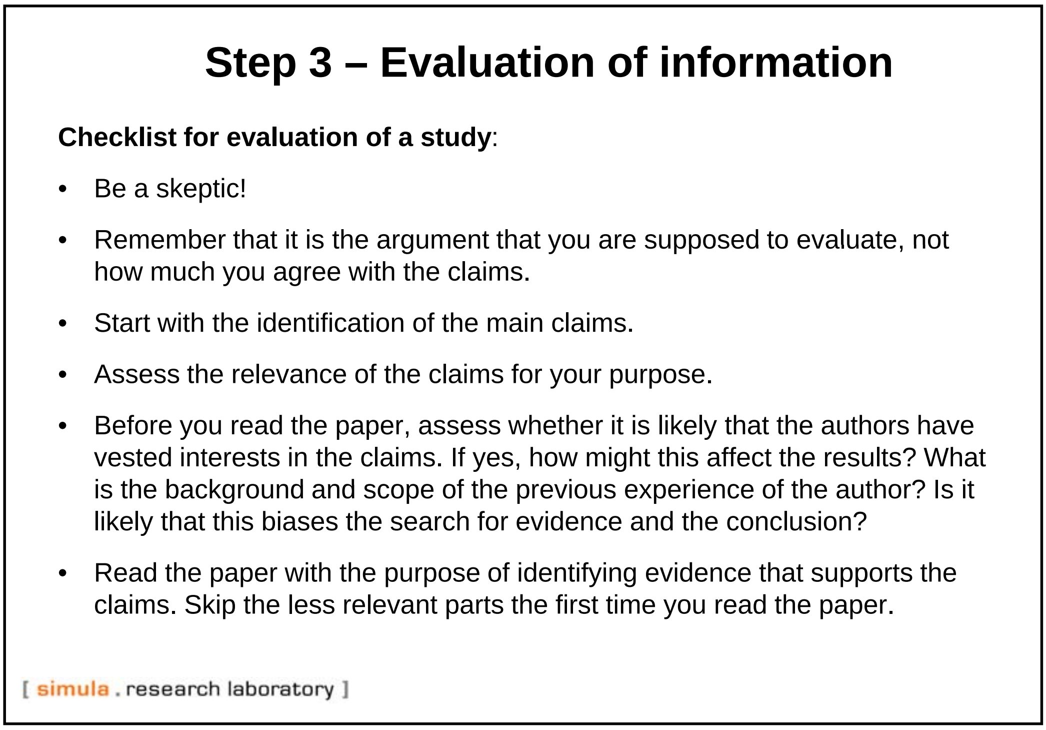
- Results (my evaluation):
  - The study findings vary very much (in itself a result).
  - Strong effect of “vested interests”. The figure below is borrowed from: *www.netc.org/openoptions/pros_cons/tco.html*.





Step 2 – Collection of information

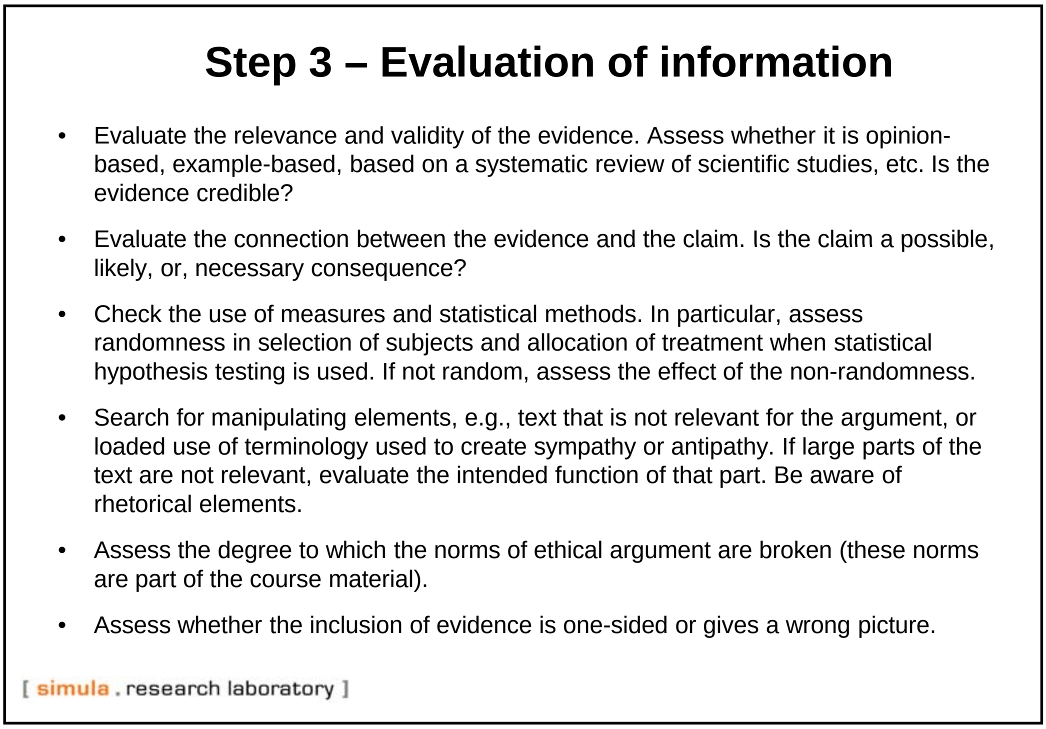
- Example of why reference clients are not of much use (in Norwegian):
  - **“***Microsoft Norge ønsker å knytte til seg flere referansekunder. Fortell oss hvordan dine forretningsmuligheter har blitt* ***styrket*** *ved hjelp av løsninger og produkter fra Microsoft, og vi forteller det videre. Som referansekunde får du ikke bare muligheten til å bli* ***profilert*** *som et selskap som tar ny og kostnadseffektiv teknologi i bruk - hvis du er raskt ute med å registrere din løsning kan du også bli med i trekningen av 10* ***gavekort****.*”
  - My translation: “Microsoft Norway wants more reference clients. Tell us about how your business opportunities has been **improved** by use of Microsoft solutions and products, and we tell it to others. As reference client you will not only have the opportunity to be **marketed** as a company that takes new and cost efficient technology in use – if you apply soon enough you will also have the opportunity to win one of 10 **present cards**”.
  - http://www.microsoft.com/norge/news/archive.mspx?year=2002
  - Why is it not likely that reference clients are valid information?



Step 3 – Evaluation of information

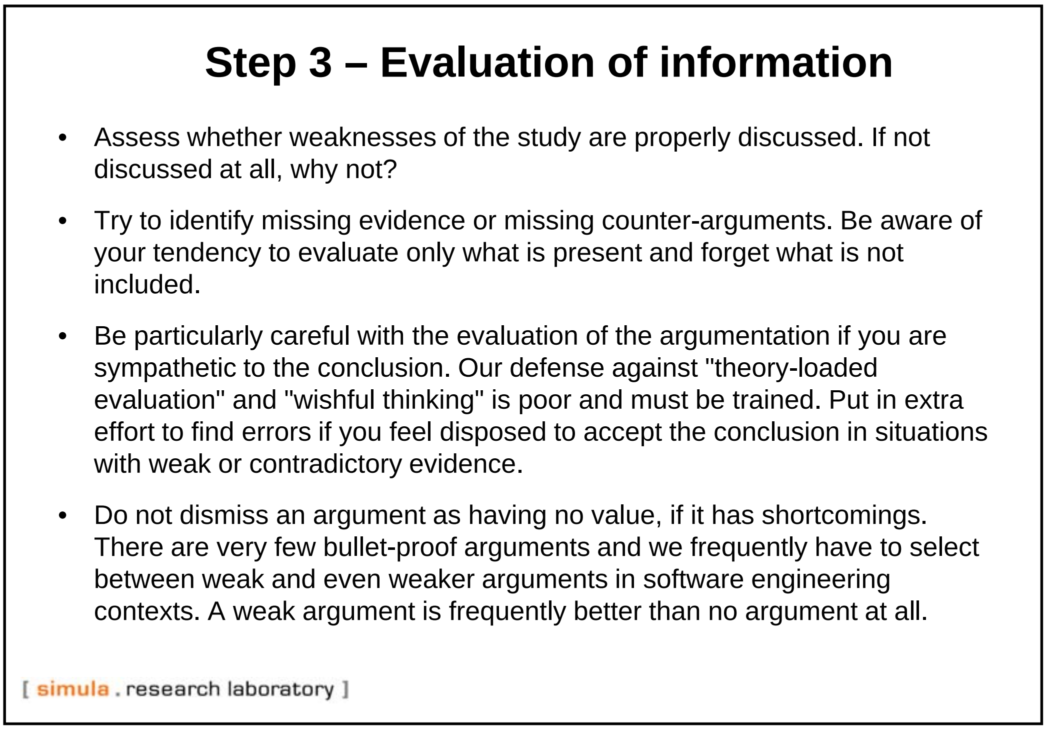
Checklist for evaluation of a study:

- Be a skeptic!
- Remember that it is the argument that you are supposed to evaluate, not how much you agree with the claims.
- Start with the identification of the main claims.
- Assess the relevance of the claims for your purpose.
- Before you read the paper, assess whether it is likely that the authors have vested interests in the claims. If yes, how might this affect the results? What is the background and scope of the previous experience of the author? Is it likely that this biases the search for evidence and the conclusion?
- Read the paper with the purpose of identifying evidence that supports the claims. Skip the less relevant parts the first time you read the paper.



Step 3 – Evaluation of information

- Evaluate the relevance and validity of the evidence. Assess whether it is opinion-based, example-based, based on a systematic review of scientific studies, etc. Is the evidence credible?
- Evaluate the connection between the evidence and the claim. Is the claim a possible, likely, or, necessary consequence?
- Check the use of measures and statistical methods. In particular, assess randomness in selection of subjects and allocation of treatment when statistical hypothesis testing is used. If not random, assess the effect of the non-randomness.
- Search for manipulating elements, e.g., text that is not relevant for the argument, or loaded use of terminology used to create sympathy or antipathy. If large parts of the text are not relevant, evaluate the intended function of that part. Be aware of rhetorical elements.
- Assess the degree to which the norms of ethical argument are broken (these norms are part of the course material).
- Assess whether the inclusion of evidence is one-sided or gives a wrong picture.



Step 3 – Evaluation of information

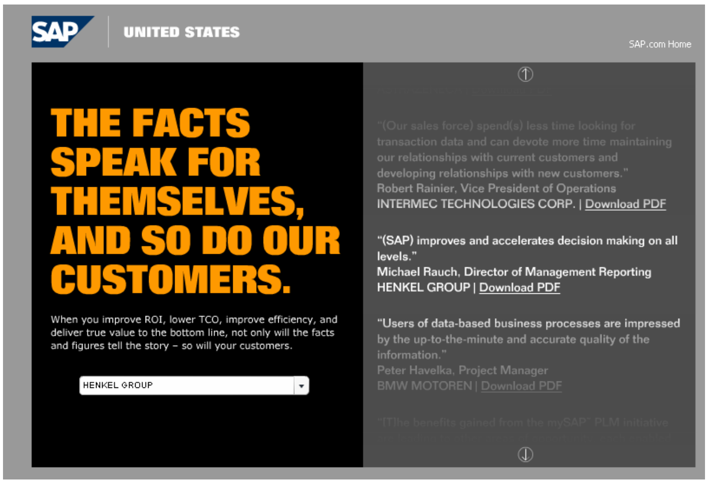
- Assess whether weaknesses of the study are properly discussed. If not discussed at all, why not?
- Try to identify missing evidence or missing counter-arguments. Be aware of your tendency to evaluate only what is present and forget what is not included.
- Be particularly careful with the evaluation of the argumentation if you are sympathetic to the conclusion. Our defense against "theory-loaded evaluation" and "wishful thinking" is poor and must be trained. Put in extra effort to find errors if you feel disposed to accept the conclusion in situations with weak or contradictory evidence.
- Do not dismiss an argument as having no value, if it has shortcomings. There are very few bullet-proof arguments and we frequently have to select between weak and even weaker arguments in software engineering contexts. A weak argument is frequently better than no argument at all.

Step 3 – Evaluation of information

- Would you trust this study?
  - “*Benchmark tests showed that SQL Server 2005 running on Windows was the most viable solution. One of the key factors influencing the technical team’s decision to choose Microsoft was the dependability of Microsoft software. The team wanted a solution that performed consistently and provided timely, reliable service.*”
  - www.microsoft.com/casestudies/casestudy.aspx?casestudyid=200945

Step 3 – Evaluate information

What do you think about these “facts”?



THE FACTS SPEAK FOR THEMSELVES, AND SO DO OUR CUSTOMERS.

When you improve ROI, lower TCO, improve efficiency, and deliver true value to the bottom line, not only will the facts and figures tell the story – so will your customers.

“(Our sales force) spend(s) less time looking for transaction data and can devote more time maintaining our relationships with current customers and developing relationships with new customers.”
Robert Rainier, Vice President of Operations
INTERMEC TECHNOLOGIES CORP. | Download PDF

“(SAP) improves and accelerates decision making on all levels.”
Michael Rauch, Director of Management Reporting
HENKEL GROUP | Download PDF

“Users of data-based business processes are impressed by the up-to-the-minute and accurate quality of the information.”
Peter Havelka, Project Manager
BMW MOTOREN | Download PDF

“[T]he benefits gained from the mySAP™ PLM initiative

Step 3 – Evaluation of information

- Sometimes weaknesses may be very difficult to identify:
  - Assume that you had read the IDC-report suggesting that Windows had lower Total Cost of Ownership (http://www.microsoft.com/windows2000/docs/TCO.pdf) than Linux.
  - The results are convincing and IDC is a serious research organization and used to completion of such studies. Their market reputation would be seriously damaged if they gave the results their clients wanted, and not the "real" ones.
  - Information about how the scenarios were chosen and how the calculations were conducted is limited and difficult to evaluate. How did this influence the evaluation?
  - BusinessWeek reports that the fairness of the evaluation may be poor:
    - "*IDC analyst Dan Kusnetzky says the company selected scenarios that would inevitably be more costly using Linux. Also, he believes Windows should be cheaper to operate, since it has been around longer, giving Microsoft more time to develop software to manage the operating system. "Microsoft has had a lot more time to work on this. I wonder why the win wasn't bigger," Kusnetzky says. Microsoft insists that it didn't rig the contest and chose the most popular uses for the software.*"
    - www.businessweek.com/magazine/content/03_09/b3822610_tc102.htm

Step 4 – Synthesis of information

- Include only essential information in the synthesis. Less important information has a tendency to remove the focus from the essential and decrease the quality of the conclusion.
- Avoid that the synthesis is a rationalization of what feels right
  - If your "gut feeling" and the analysis diverge, follow the analysis (unless your own satisfaction with the choice is not of great importance)
- The reports I read on Microsoft vs Linux can be summarized as follows:
  - There seem to be no LARGE systematic, well-documented differences in TCO between Linux and Windows. If any, it seems that Windows has had lower TCO – but this may easily change with more users of Linux.
  - There is a striking lack of studies not paid by one of the parties (Windows or Linux-proponents). A few studies seem, however, to have a proper research methods.
  - Conclusion: The uncertainty/variation in results is so high that the organization cannot emphasize these differences in their choice between Linux and Windows (given that valid studies are relevant for the organization's own context). Other criteria should consequently be emphasized.