[simula . research laboratory]

Research on Software Engineering Research

Dag Sjøberg

Professor, Department of Informatics, University of Oslo/
Software Engineering, Simula

The Software Engineering Research Method Project at Simula

The overall purpose is to support the private and public services and industries in developing better IT-systems using fewer resources by

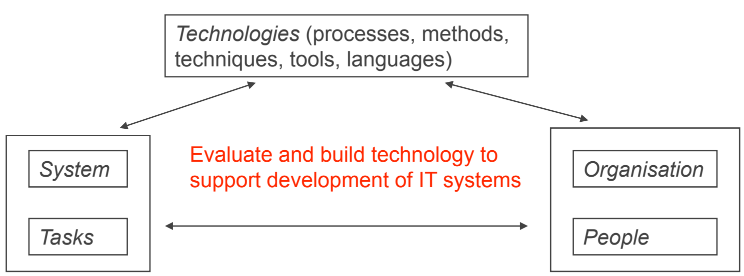
(1) advancing the state of the art of empirical software engineering research by developing new, or improving existing, methods and infrastructures for conducting empirical studies in software engineering, and

(2) proposing and validating theories on the basis of the studies primarily carried out in software development organizations.

Purpose of doing research on research

- Help a research community to define the goals of its research
- Help a research community to define quality standards for the way the research is carried out
- Help a research community to achieve these quality standards

Purpose of SE research: Improve the way software systems are built and maintained



Software Engineering Research

SE research is about

(1) the *development* of new, or *modification* of existing, technologies to support SE activities

(2) the *evaluation* and *comparison* of the effect of using such technology in the often very complex interaction of individuals, teams, projects and organisations, and various types of tasks and software systems. This may be referred to as ***evidence-based*** or ***empirical software engineering***, whose goal is to:

- to enable the development of scientific knowledge or evidence about how useful different SE technologies are in which situations. Such knowledge or evidence should guide the development of new SE technologies and be a major input to important SE decisions in industry.

Scientific Evaluation of SE Technology

- Today: Mostly based on anecdotal evidence, personal opinion, arbitrary tests, etc.
- Sciences that study real-world phenomena use empirical methods by necessity, which involve systematic observation and experimenting, rather than deductive logic or mathematics
- If SE research is to be a science, it must include the use of empirical methods

Examples

- Pair programming, and its interaction with individual skills and system complexity
- Effect of UML in the context of software evolution
- Regression testing: test selection, minimization, prioritization

Challenges

- More empirical studies
- Higher quality studies
  - More relevant studies
  - More valid studies
    - Internal validity*
    - Construct validity*
    - External validity*
    - Statistical conclusion validity*
    - Reliability (experiments, case studies, etc.)#
- More focus on synthesizing evidence
- Theory building

* Shadish, W.R., Cook, T.D. and Campbell, D.T., Experimental and Quasi-Experimental Designs for Generalized Causal Inference. Boston: Houghton Mifflin, 2002, #Yin, 2003

Current SE research literature

- Percentage of articles that report empirical studies :
  - Tichy: 17%
  - Glass et al.: 14%
  - Sjøberg et al.: 12-17%
- Primary studies
  - Controlled experiments 1.9% (Sjøberg et al.)
  - (Personal opinion) Surveys 1.6% (Glass *et al*.)
  - Case studies 12% (Holt)
  - Action research 0% (Glass)
- Reviews and meta-analysis: 1-3% of papers
- **Rough estimate: 180 studies a year**

Need: ~2000 studies

- Assume there are 1000 research questions of high industrial importance that are meaningful to decide empirically, and
- assume that a research question requires at least 20 high quality studies, conducted over the last 10 years.
- This requires that we conduct at least 2000 high-quality empirical studies every year.

See more details in: D.I.K. Sjøberg, T. Dybå and M. Jørgensen. The Future of Empirical Methods in Software Engineering Research, In Future of Software Engineering (FOSE '07), edited by Briand L. and Wolf A., Minneapolis, US, 23-25 May 2007. IEEE-CS Press, pages 358-378, 2007.

More empirical studies

State of Practice	Target (2020-2025)
Relatively few empirical studies in SE research. Focus on developing new technology	Large number of studies covering all important fields of SE and using different empirical methods. Most research that leads to new or modified technology is subject to empirical evaluation
Empirical methods not part of industrial practice	Most large software development organizations conduct empirical studies as part of decisions making and process improvement

More relevant studies

- Relevance on topic
- Relevance of applicability of the results, see external validity later

More relevant studies

Why is relevant topic important?

"Currently, research priorities in the IS field seem to be driven more by the interests of researchers rather than by the needs of practice or society. Hirschheim et al. (1996) see this as a good thing, in that it encourages diversity and promotes academic freedom. However, in an applied discipline, it also reflects a lack of social accountability. For example, there would be a public outcry if medical researchers spent their time researching health problems that interested them while ignoring the major health problems in society."

[Daniel L. Moody, Proceedings of the twenty first international conference on Information systems Brisbane, Australia, pp. 351-360, 2000]

More relevant studies

State of Practice	Target (2020-2025)
Few results answer questions posed by industrial users, e.g., "Which method should we use in our context?" Current focus is on comparing mean values of technologies without a proper understanding of individual differences or the studied population	More focus on individualized results, individual differences, and better descriptions of populations and contexts; why, when and how technology X is better than technology
Reference points for comparisons of technologies are frequently not stated, or not relevant	New technology is compared with relevant alternative technology used in the software industry
One may question the industrial relevance of many SE studies	More case studies and action research. Experiments should show more realism regarding subjects, technology, tasks, and software systems

More Valid Studies

Internal validity

The *internal validity* of an experiment is “the validity of inferences about whether observed co-variation between *A* (the presumed treatment) and *B* (the presumed outcome) reflects a causal relationship from *A* to *B* as those variables were manipulated or measured” [Shadish, 2002]. Changes in *B* may have alternative causes than the manipulation of *A*. An alternative cause for the outcome is a *confounding factor*.

Internal validity

Systematic review

- Randomized allocation of treatments to subjects is one way of handling threats to internal validity. Done in 58% of experiments [Kampenes et al. 2007].
- Randomization is not always desirable or possible in SE, hence 35% quasi-experiments.
- Only half of the quasi-experiments measured a pretest score to control for selection bias, and only 8% reported a threat of selection bias.

State of Practice	Target (2020-2025)
Results are often not robust due to lack of replications and reliance on only one type of research design.	Replications and triangulation of research designs are frequently used.

More Valid Studies

Construct validity

- We need to measure something to understand it, but just as importantly, we need to understand something in order to measure it
- For example: Quality = number of errors? What about functionality, usability, maintainability, etc. And what kind of errors, found where, found when? Compared with what?
- In general, low construct validity in SE studies, although little systematic investigation on this issue. Simula plans to carry out a systematic review in this area

More Valid Studies

External validity – Generalisation

The validity of inference about whether the cause-effect relationship holds over variation in:

- Actors: individual, teams, project,organisation or industry
- Technology: process model, method, technique, tool or language
- Activities: plan, create, modify or analyze (a software system)
- Software systems: many dimensions, such as size, complexity, application domain, business/scientific/student project or administrative/embedded/real time, etc.

Dimensions of Generalization

	Statistical generalization	Analytical generalization
Individual studies	Statistical hypothesis testing	Generalization through theory or analogy
Collection of studies	Meta analysis	Research synthesis, aggregation of evidence, and theory

Generalization

State of Practice	Target (2020-2025)
The scope of validity of empirical studies is rarely defined explicitly	The scope is systematically and explicitly defined and reported
Statistics-based generalization is the dominant means of generalization	Studies include a diverse and reflected view on how to generalize, particularly through the use of theory

[simula . research laboratory]

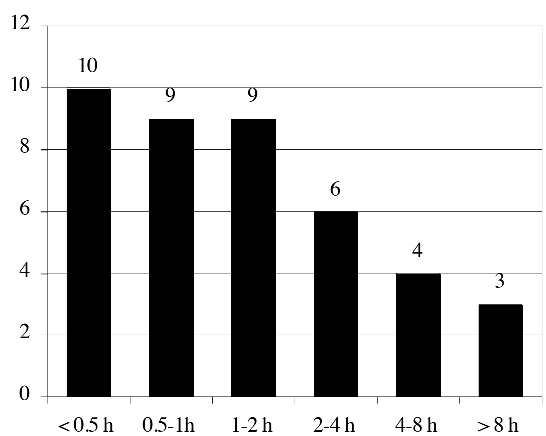
Challenge

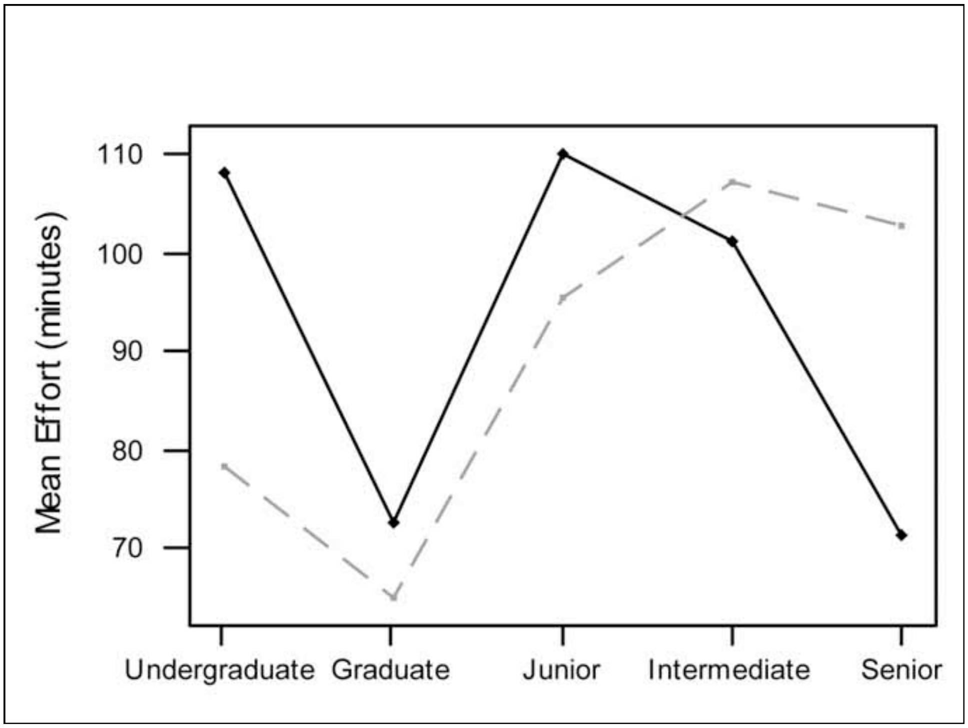
- The applicability of the experimental results to industrial practices is in most cases hampered by the experiments' lack of *realism* and *scale* regarding, that is, the challenge of achieving external validity
- Many aspects of the complexity of software engineering only manifest themselves in controlled experiments if the experiments involve a sufficiently large number of subjects, tasks and systems, for example, differences among subgroups of subjects
- How to generalise from SE experiments. How do we convince practitioners and managers in industry that the results are relevant to them?

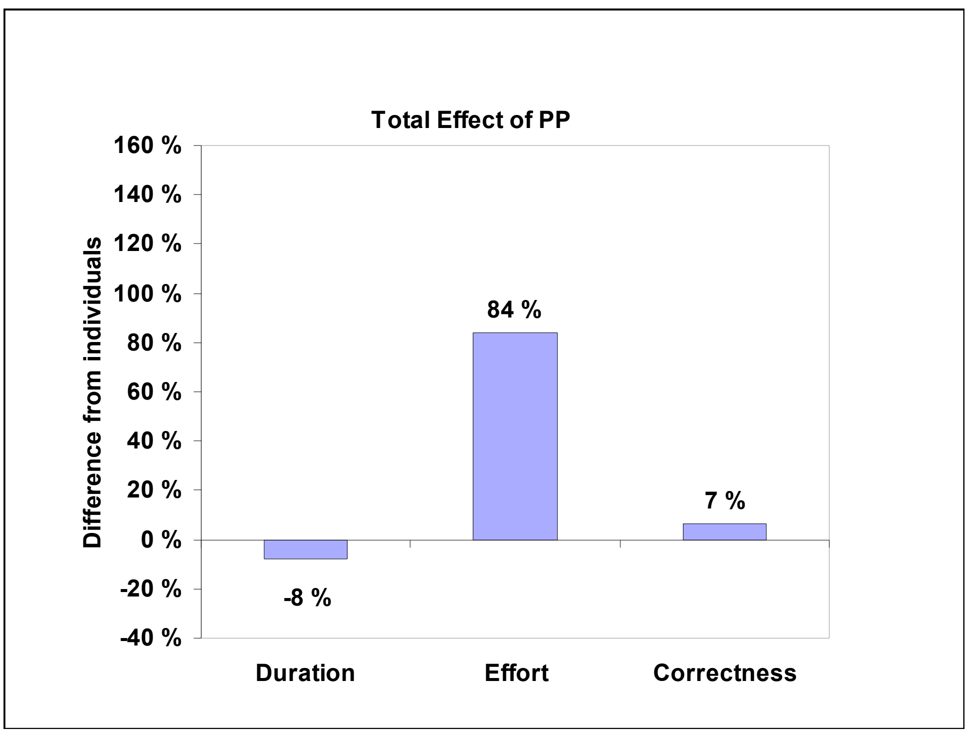
[simula . research laboratory]

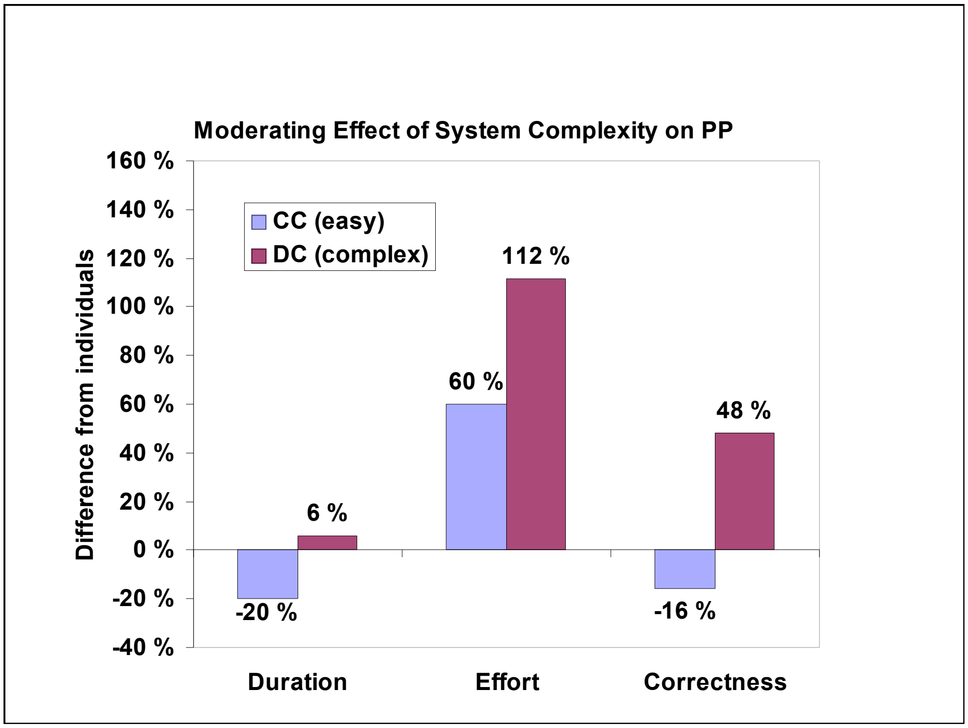
Development tasks in industry usually take longer and are more complex than in most experiments:

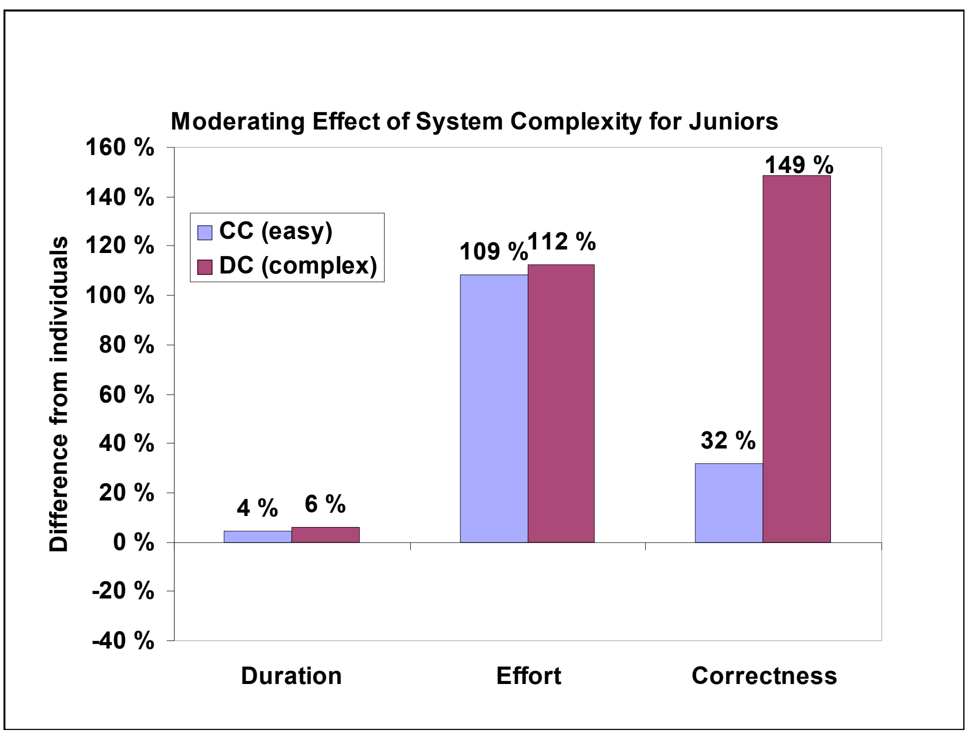
Duration of experiments with time measurements

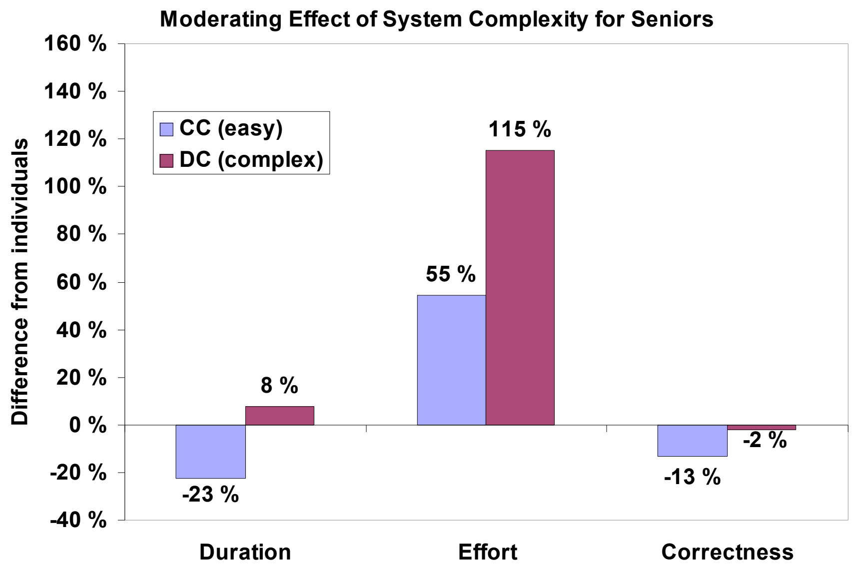



Mean Effort (minutes)
110
100
90
80
70
Undergraduate
Graduate
Junior
Intermediate
Senior


Total Effect of PP
Difference from individuals
160 %
140 %
120 %
100 %
80 %
60 %
40 %
20 %
0 %
-20 %
-40 %
84 %
7 %
-8 %
Duration
Effort
Correctness


Moderating Effect of System Complexity on PP
160 %
140 %
120 %
100 %
80 %
60 %
40 %
20 %
0 %
-20 %
-40 %
Difference from individuals
CC (easy)
DC (complex)
-20 %
6 %
60 %
112 %
-16 %
48 %
Duration
Effort
Correctness


Moderating Effect of System Complexity for Juniors
160 %
140 %
120 %
100 %
80 %
60 %
40 %
20 %
0 %
-20 %
-40 %
Difference from individuals
CC (easy)
DC (complex)
4 %
6 %
109 %
112 %
32 %
149 %
Duration
Effort
Correctness



Is a helicopter better than a bike?

The effect of PP “depends on”

Programmer expertise	Task complexity	Use PP?	Comments
Junior	Easy	Yes	Provided that increased quality is the main goal
	Complex	Yes	Provided that increased quality is the main goal
Intermediate	Easy	No	
	Complex	Yes	Provided that increased quality is the main goal
Senior	Easy	No	
	Complex	No*	

* Unless you are sure that the task is too complex to be solved satisfactorily even by solo seniors

- The performance of the various categories may depend on their relevant education, work experience, the actual task and system, development technology, etc.
- In the survey of 113 experiments, 7 involved both students and professionals. Only 3 measured difference in performance: partly no difference, partly professionals better.

Subjects

Subject Category	Reported Subject Types	N	%
Undergraduates	Undergraduates , Bachelors , Third and fourth -year students, Last-year students, Honors and Majors .	2969	54.1
Graduates	Graduate students , Students following graduate courses or Master’s programs , MSc and PhD students .	594	10.8
Students, type unknown	Students in computer science, Students .	1203	21.9
Professionals	Developers, Practitioners, Software engineers, Analysts , Domain experts, Business managers , Facilitators , Professionals.	517	9.4
Scientists	Professors, Post-doctorates , Staff members of educational institutions .	74	1.3
Unknown		131	2.3
Total		5488	100

Sjøberg, D.I.K., Hannay, J.E., Hansen, O., Kampenes, V.B., Karahasanović, A., Liborg, N.-K. and Rekdal, A.C. A Survey of Controlled Experiments in Software Engineering, IEEE Transactions on Software Engineering, 31(9): 733–753, 2005

More Valid Studies

Statistical conclusion validity

The validity of inferences about the correlation (covariation) between treatment and outcome.

- Statistical power is the probability that a statistical test will correctly reject the null hypothesis. A test without sufficient statistical power will not provide enough information to draw conclusions regarding the acceptance or rejection of the null hypothesis.
- An effect size quantifies the effects of an experimental treatment. Whereas *p*-values reveal whether a finding is *statistically* significant, effect size indicates *practical* significance, importance, or meaningfulness.

Why that many subjects? Power analysis

Research question:

What is the effect regarding duration, effort and correctness of pair programming for various levels of system complexity and programmer expertise when performing change tasks?

- 2x2x3 fixed-effect analysis of covariance:
  pair programming (two levels), control style (two levels) and expertise (three levels), resulting in twelve levels/groups
- N = 170 (85 individuals and 85 pairs)
- N = 14 in each of the 12 groups

Statistical Conclusion Validity

State of Practice	Target (2020-2025)
Stat. methods are used mechanically, with little focus on limitations and assumptions. Populations not defined, and for experiments, lack of power analysis and effect size estimation.	The use of statistical methods is mature. Populations are well defined, and power analysis and effect size estimation are conducted when appropriate.

More Repeatable Studies

Can the study be repeated with the same results?

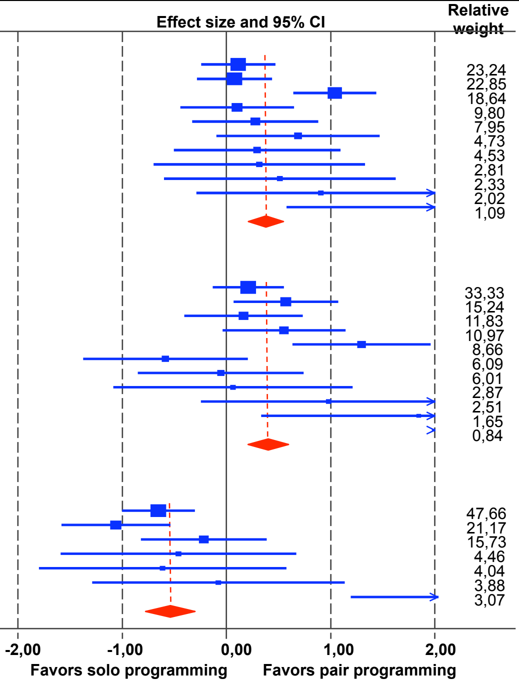
Repeatability of a study represents the ability of other investigators to follow the same procedures, to perform exactly the same study, and to arrive at the same findings and conclusions (Yin 2003).
(Yin uses the term "reliability" instead of "repeatability")

Synthesizing Evidence

- *Primary*: collection and analysis of data
  - Experiments, surveys, case studies, action research, and others
- *Secondary:* research synthesis, summary, integration and combination of the findings of different primary research studies on a certain topic
  - Systematic reviews (see lecture 16.10.2007), meta-analysis

    Reviews on research methods (= PhD of Vigdis By Kampenes):
    - A survey of controlled experiments in software engineering
    - A systematic review of statistical power in software engineering experiments
    - A Systematic review of effect size in software engineering experiments
    - A systematic review of quasi-experiments in software engineering

Study name	Effect size	Lower limit	Upper limit	Relative weight
a) Quality				
P07a	0,11	-0,24	0,46	23,24
S06a	0,08	-0,28	0,43	22,85
S00	1,04	0,65	1,43	18,64
S03	0,10	-0,44	0,64	9,80
S05b	0,28	-0,32	0,88	7,95
P07b	0,69	-0,09	1,46	4,73
S02	0,30	-0,50	1,09	4,53
S06c	0,32	-0,69	1,32	2,81
S06b	0,51	-0,59	1,62	2,33
P98	0,91	-0,28	2,10	2,02
S06d	2,20	0,58	3,82	1,09
Overall effect	**0,38**	**0,21**	**0,55**	
b) Duration				
P07a	0,21	-0,13	0,54	33,33
S05a	0,57	0,07	1,07	15,24
S01	0,16	-0,40	0,73	11,83
S03	0,55	-0,03	1,14	10,97
S05b	1,30	0,64	1,95	8,66
P07b	-0,59	-1,37	0,20	6,09
S02	-0,05	-0,84	0,74	6,01
P02	0,06	-1,08	1,21	2,87
S06b	0,98	-0,24	2,20	2,51
S06d	1,85	0,34	3,35	1,65
P98	4,09	1,98	6,20	0,84
Overall effect	**0,40**	**0,21**	**0,59**	
c) Effort				
P07a	-0,68	-1,03	-0,34	47,66
S05a	-1,09	-1,61	-0,58	21,17
S05b	-0,25	-0,85	0,35	15,73
S05c	-0,49	-1,62	0,63	4,46
S06b	-0,64	-1,83	0,54	4,04
S06d	-0,11	-1,31	1,10	3,88
S06c	2,52	1,17	3,88	3,07
Overall effect	**-0,57**	**-0,81**	**-0,33**	



Synthesis of evidence

State of Practice	Target (2020-2025)
Narrative, biased reviews and little appreciation of the value of systematic reviews	Scientific methods are used to undertake integrative and interpretive reviews to inform research and practice
The number and coverage of systematic reviews is very limited	Policy-makers, practitioners, and the general public have up-to-date and relevant systematic reviews and evidence-based guidelines and checklists at their disposal
Lack of common terminology and appropriate descriptors and keywords	The SE community is mature regarding understanding and use of basic terminology, descriptors and keywords. The electronic resources have high quality in their support of information about SE research
No common understanding of SE phenomena	Agreed-upon conceptual and operational definitions of key SE constructs and variables
Limited advice on how to combine data from diverse study types	Methods are available for synthesizing evidence from a variety of perspectives and approaches to research and practice

Theory building

State of Practice	Target (2020-2025)
Generally, little use of theories. The theories used mainly justify research questions and hypotheses; some explain results; very few test or modify theory	Most SE studies involve theories. Considering using, testing, modifying or formulating theory is part of any empirical work
Almost no SE-specific theories are proposed	Many SE theories are proposed and tested
Theories are generally poorly documented	There are widely used standards for describing theories in a clear and precise way
Difficult to identify the theories that actually are used or have been proposed	For each SE sub-discipline, there are web-sites and systematic reviews that systematize and characterise relevant theories

How to improve the quality of SE research?

- Increasing competence regarding how to conduct empirical studies
  - Guidelines and empirical methods included in SE curricula
  - Develop infrastructures to support the conducting of studies
- Improving the links between academia and industry
  - Get involved in SPI work in companies
  - Give seminars and courses where studies are included
- Developing common research agendas
  - More concentrated effort – SE researchers should work on common research programs
- Increasing the resources available

Increasing competence

State of Practice	Target (2020-2025)
Researchers often do not build sufficiently on previous research results	There is a strong emphasis on building on previous research results, including those from other disciplines
Skills in conducting controlled experiments and reviews have improved, but not skills in conducting surveys, case-studies and action research	Research method and design elements are carefully selected and combined, based on an in-depth understanding of their strengths and weaknesses

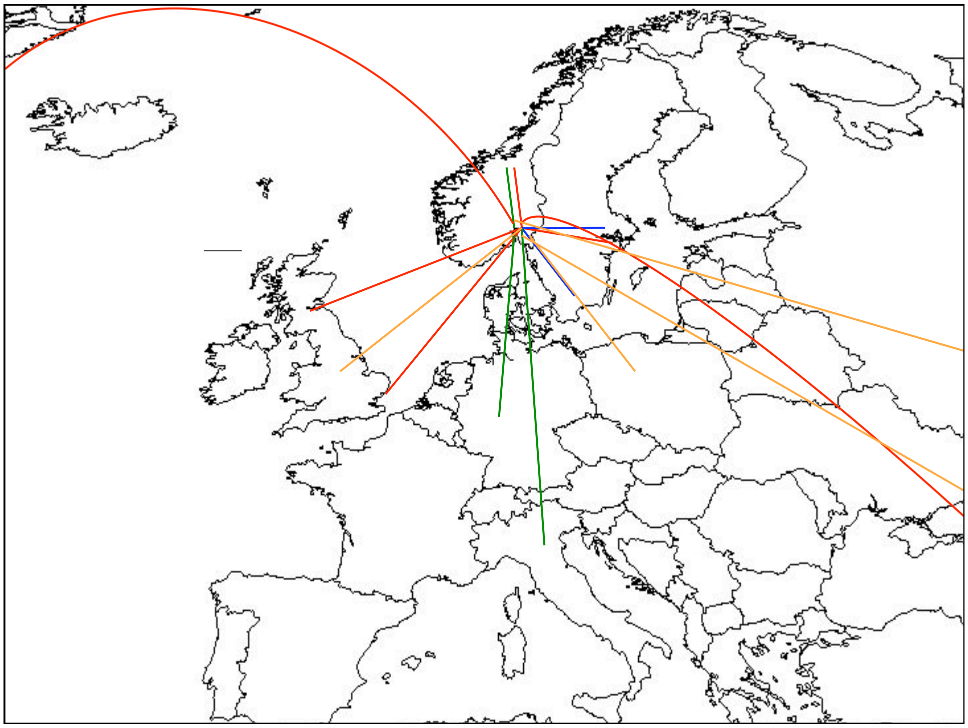
Consulting related disciplines

Software engineering is typically performed by humans in organisations. Hence, Simula has established research collaborations with disciplines such as psychology, sociology and management, in addition to statistics.

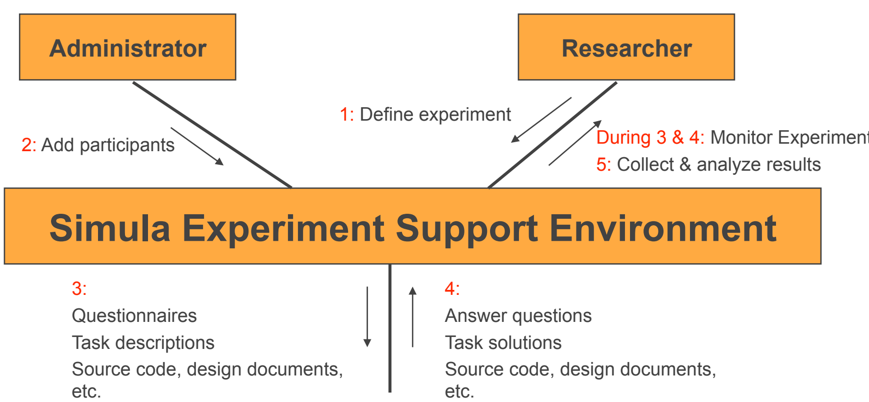
Need for infrastructures, f.ex.:

The logistics of controlled experiments is work intensive and error prone

- Personal information and background information of subjects must be collected
- General information and specific task documents must be printed and distributed
- Solution documents must be collected



Web-based tool support (SESE)



Key functionality of SESE

- real-time monitoring of the experiment
- flexibility of defining new kinds of questions and measurement scales
- automatic recovery of experiment sessions
- automatic backup of experimental data
- multi-platform support for downloading experimental materials and uploading task solutions

SESE is built on top of a commercial human resource management system, and is partly being developed by an external company

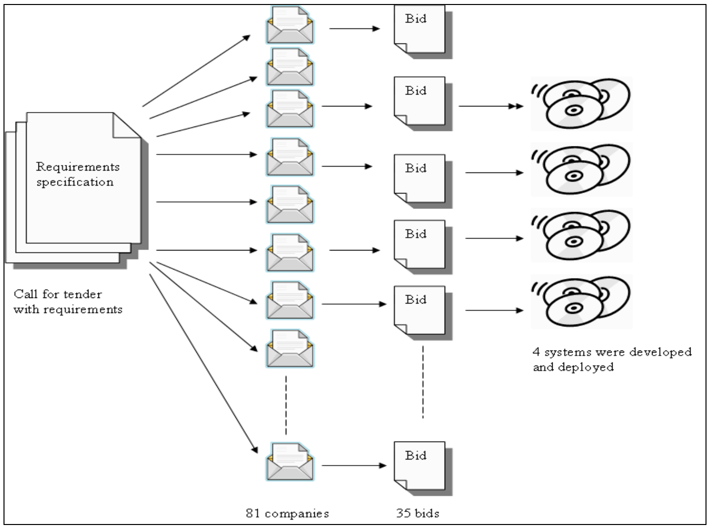
[E. Arisholm, D. I. Sjøberg, G. J. Carelius and Y. Lindsjørn. A Web-based Support Environment for Software Engineering Experiments, Nordic Journal of Computing 9(4):231-247, 2002.]

Practical organisation of large experiments

- Ask for a local project manager of the company who selects subjects according to the specification of the researchers, ensures that the subjects actually turn up, ensures that the necessary tools are installed on the PCs, and carries out all other logistics, accounting, etc.
- Motivate the experiment up-front: inform the subjects about the purpose of the experiment (at a general level) and the procedure (when to take lunch or breaks, that phone calls and other interruptions should be avoided, etc.).
- Ensure that the subjects do not talk with one another in breaks, lunch, etc.
- Ensure the subjects that the information about their performance is kept confidential (both within company and outside).
- Ensure the company that its general performance is kept confidential.
- Monitor the experiment, that is, be visible and accessible for questions.
- Give all subjects a small training exercise to ensure that the PC and tool environment are working properly.
- Ensure the company and subjects that they will be informed about the results of the experiment.
- Provide a proper experiment support environment that is used to set up and monitor the experiment, and collect and manage the experimental data.

[simula . research laboratory]

A study of reproducibility in SE



[simula . research laboratory]

Developing common research agendas

- Common research agendas should be established to improve empirical work per se, but also for specific SE topics, for example, distributed software development.
- A more ambitious, long-term goal would be to establish a program in software engineering similar to the Human Genome Project and CERN.

The costs of running large experiments with professionals

- "The experimental approach is not without limits. First of all, the costs are high and in some cases may become prohibitive. It is clearly impossible to do an experiment with hundreds of professionals, so smaller groups or case studies will have to suffice."

  [A. Endres and D. Rombach, A Handbook of Software and Systems Engineering. Empirical Observations, Laws and Theories, Fraunhofer IESE Series on Software Engineering. Pearson Education Limited, 2003]

- "practitioners are understandably skeptical of results acquired from a study of 18-year-old college freshmen."

  "finding 100 developers willing to participate in such an experiment is neither cheap nor easy. … But even if a researcher has the money, where do they find that many programmers?"

  [W. Harrison, "Skinner Wasn't a Software Engineer", Editorial, *IEEE Software,* May/June, 2005]

Examples of experiments at Simula

- 99 consultants from 8 companies
  - one-day experiment that compared two different object-oriented control styles
- 295 consultants from 29 companies in Norway, Sweden and the UK
  - one-day experiment that tested the effect of pair programming
- 39 consultants from 11 companies
  - Three-day experiment on design patterns
- 20 programmers from 13 companies
  - worked individually from one to two weeks in an experiment on UML
- 35 companies presented bids for a web-based system that we needed
  - 4 were selected to actually build the system independently of each other.
  - The teams (2-3 developers from each company) spent from 7 to 25 person-weeks each
- 30 companies from 11 countries in Europe and Asia presented their bids.
  - 4 companies built the system
  - each spent from 10 to 20 person-weeks
- 40 companies from 8 countries in Europe and Asia estimated five projects each
  - The spent 2-4 person-weeks each

How do we get the subjects? – Hire consultants

- The experiments listed above cost between €50,000 and €230,000.
- We paid the companies ordinary consultancy fees for individuals or fixed price for a whole project, like any other ordinary customer.
  - The companies have routines for defining (small) projects with local project management, resource allocation, budgeting, invoicing, providing satisfactory equipment, etc.
- Difficult to find many experiment subjects employed in an in-house software development company because the management will typically prioritize the next release of their product.

Large-scale empirical work requires a great amount of resources

- At Simula we have used about 25% of budget for experiments, mainly at the expense of more researchers.
- In research grants applications, one budgets for money for positions, equipment and travel; why not include money for experiments?
- SE researchers should contribute to making the development of software systems a mature industry. Given the importance of software systems in society, there is no reason why research projects in SE should be less comprehensive and cost less than large projects in other disciplines, such as physics and medicine. The U.S. funding for the Human Genome Project was $437 million over 16 years. If many scientific activities related to genomics are included, the total cost rises to $3 billion! CERN's annual budget is about $800 million.