The Simula approach to experimentation in software engineering

Dag Sjøberg
History of Simula Research Laboratory

1991: Decision to close the airport at Fornebu, Oslo
1991-1997: Political debate concerning possible use of the old airport
1999: Funding approved for a research institute at Fornebu
2000: The Parliament decides that IT-Fornebu shall develop a Knowledge Park at the old airport
2000: Three research groups selected on basis of applications from 17 Norwegian university groups
Simula established in 2001

- 83 employees
- Shareholding company (Norwegian state 80 %, Sintef+Norwegian computing centre = 20 %)
- Research departments:
  - Networks and Distributed Systems
  - Scientific Computing
  - Software Engineering
- Simula Innovation
How is Simula different from a (Norwegian?) university?

• More focused on research
  – large and focused long term research projects

• Stronger management of research directions
  – less individual freedom on topics

• Stronger emphasis on industrial relevance of the research
The Software Engineering Department

<table>
<thead>
<tr>
<th>Full time employees</th>
<th>Part time employees</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  Research director (Prof. Dag Sjøberg)</td>
<td>1  Visiting professor (Lionel Briand, Carleton University)</td>
</tr>
<tr>
<td>1  Professor (Magne Jørgensen)</td>
<td>2  Visiting researchers (Tore Dybå, SINTEF; Marek Vokác, SuperOffice)</td>
</tr>
<tr>
<td>1  Assoc. professor (Erik Arisholm)</td>
<td></td>
</tr>
<tr>
<td>3  Research scientists/post-docs (Bente Anda, Amela Karahasanovic, Jo Hannay, Kjetil Moløkken-Østvold)</td>
<td></td>
</tr>
<tr>
<td>1  Knowledge manager (Hans Christian Benestad)</td>
<td></td>
</tr>
<tr>
<td>7  PhD students (James Dzidek, Hans Gallis, Stein Grimstad, Vigdis By Kampenes, Tanja Gruske, Gunnar Bergersen, Hans Christian Benestad)</td>
<td></td>
</tr>
</tbody>
</table>

Three groups:
- Estimation of Software Tasks (Magne Jørgensen)
- Object-Oriented Analysis and Design (Erik Arisholm)
- Software Engineering Research Methods (Dag Sjøberg)
Goal of software engineering research

The ultimate goal of the research in software engineering is to support the private and public software industry in developing higher quality systems with improved timeliness in a more cost-effective and predictable way.
Is a helicopter better than a bike?
Purpose of SE research: Improve the way software systems are built and maintained

- Technologies (processes, methods, techniques, tools, languages)
- System
- Tasks
- Organisation
- People

Evaluate and build technology to support development of IT systems

- There are often hundreds of alternative technologies: How should the industry (and others who build software) judge what technologies (processes, methods, techniques, guidelines, and tools) are useful for different kinds of developer, performing different kinds of task, on different kinds of system, in different kinds of organisation?
- Many achievements have been made in the empirical SE community, but we are still far from generally able to answer this question.
Controlled experiments

One contribution of the empirical SE community is the conducting of experiments to evaluate and compare industrial software engineering technologies.
Challenge

- How do we convince practitioners and managers in industry that the results of controlled experiments are relevant to them?

- The applicability of the experimental results to industrial practices is in most cases hampered by the experiments’ lack of realism and scale regarding subjects, tasks, systems and environments, that is, the challenge of achieving external validity.
State of the art in SE experimentation


<table>
<thead>
<tr>
<th>Journal/Conference</th>
<th>Total no. of articles investigated</th>
<th>Articles reporting controlled experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>EMSE</td>
<td>124</td>
<td>22</td>
</tr>
<tr>
<td>ISESE</td>
<td>20</td>
<td>3</td>
</tr>
<tr>
<td>METRICS</td>
<td>177</td>
<td>10</td>
</tr>
<tr>
<td>JSS</td>
<td>886</td>
<td>24</td>
</tr>
<tr>
<td>TSE</td>
<td>687</td>
<td>17</td>
</tr>
<tr>
<td>ICSE</td>
<td>520</td>
<td>12</td>
</tr>
<tr>
<td>IST</td>
<td>745</td>
<td>8</td>
</tr>
<tr>
<td>SME</td>
<td>186</td>
<td>2</td>
</tr>
<tr>
<td>IEEE SW</td>
<td>532</td>
<td>4</td>
</tr>
<tr>
<td>TOSEM</td>
<td>125</td>
<td>1</td>
</tr>
<tr>
<td>IEEE Comp</td>
<td>780</td>
<td>0</td>
</tr>
<tr>
<td>SP&amp;E</td>
<td>671</td>
<td>0</td>
</tr>
<tr>
<td>All</td>
<td>5453</td>
<td>103</td>
</tr>
</tbody>
</table>
### Subjects

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

© Dag Sjøberg, EASE’2006, Keele, 10 April 2006
Realism (representativeness) of tasks, systems and environments

- A grand challenge in SE experimentation is how we generalise from the specific tasks, systems and environments of SE experiments.

- Not aware of suitable taxonomy or classification of these aspects for SE.

- Nevertheless, development tasks in industry usually take longer and are often more complex than is the case in most experiments.
Duration of experiments with time measurements
Why is scale important?

• Easier to obtain a representative sample of the target population.
  – One of 113 experiments reported sampling from a well-defined target population.

• Many aspects of the complexity of software engineering only manifest themselves in controlled experiments if the experiments involve a sufficiently large number of subjects and tasks, for example, differences among subgroups of subjects
Example of difference among subgroups

[Graph showing mean effort (minutes) over time for Undergraduate, Graduate, Junior, Intermediate, and Senior categories. The graph compares two designs: DC and CC.]
Another example, (quasi) experiment on pair programming

• 295 junior, intermediate and senior professional Java consultants from 29 companies were paid to participate (one work day)
• 99 individuals (conducted in 2001/2002)
• 98 pairs (conducted in 2004/2005)
  – Norway: 41
  – Sweden: 28
  – UK: 29
• The pairs and individuals performed the same Java maintenance tasks on either:
  – a "simple" system (centralised style) or
  – a "complex" system (delegated style)
• We measured duration (elapsed time), effort (cost) and correctness of their solutions
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
Junior developers benefited most from pair programming, particularly on the complex system

- The benefits of PP are reduced with increasing *programming skills* of the individuals
- The benefits of PP are reduced with decreasing *task complexity*
- Pair programming requires significantly more effort than individual programming (regardless of programmer category)
- **Juniors:**
  - PP significantly improves correctness
  - The effect of PP on *correctness* is highest for the tasks based on the delegated control style
  - Junior pairs obtain almost the same correctness as intermediate and senior pairs (≈ 80 percent correct)
- **Seniors:**
  - No clear benefits of pair programming
Total Effect of PP

Difference from individuals

Duration
Effort
Correctness

-8 % 84 % 7 %
Moderating Effect of System Complexity on PP

- CC (easy) vs. DC (complex)

- Duration: 6%
- Effort: 60%
- Correctness: 48%
Moderating Effect of System Complexity for Juniors

Difference from individuals

- Duration
- Effort
- Correctness

CC (easy)
DC (complex)

4% 6%
109% 112%
32%
149%
Moderating Effect of System Complexity for Seniors

- Duration: -23%
- Effort: 55%
- Correctness: -13%

CC (easy)
DC (complex)
Characterising the subject population

- The results indicate that there may be significant difference between categories of subjects.
- The performance of the various categories may depend on their relevant education, work experience, the actual task and system, development technology, etc.
- Characterising the level of expertise of a subject is difficult (pre-test best alternative?), but necessary to judge the populations to which the results can be generalised.
- In the survey of 113 experiments, 7 involved both students and professionals. Only 3 measured difference in performance: partly no difference, partly professionals better.
”Impossible” to run 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.”


• “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?”

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
800 consultants from 60 companies (last 5 years)
How do we get the subjects?

• In all these experiments, we paid the companies ordinary consultancy fees for individuals or fixed price for a whole project

• The experiments listed above cost between €50,000 and €230,000
Hiring consultants

- Difficult to find many experiment subjects employed in an in-house software development company because the management will prioritize the next release of their product.

- However, most IT consultancy companies don’t have such constraints and are contracted to carry out various kinds of assignments.

- We buy a service from consultancy companies like any other ordinary customer.
  - They have routines for defining (small) projects with local project management, resource allocation, budgeting, invoicing, providing satisfactory equipment, etc.
How do we get the money?

• At Simula, research and administrative leader is the same person.
• We have relatively few constraints on how we spend the money as long as we can envisage a good research outcome.
• Decided to use 25% of budget for experiments, mainly at the expense of employing a larger number of researchers.
Apply for money to conduct experiments

- Finding the money to fund comprehensive experiments is a matter of politics. I have yet to meet a researcher who has applied to funding bodies for money to pay for professionals to take part in experiments; neither have I.

- In research grants applications, we budget for money for positions, equipment and travel; why not include money for experiments?

- Compared with large projects in other disciplines, e.g., physics and medicine, we are talking about a relatively small amount of money
## Reward mechanisms in SE experiments

<table>
<thead>
<tr>
<th>Reward</th>
<th>Experiment</th>
<th>%</th>
<th>Participant</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grade</td>
<td>10</td>
<td>8.8</td>
<td>732</td>
<td>13.3</td>
</tr>
<tr>
<td>Extra credits</td>
<td>9</td>
<td>8.0</td>
<td>660</td>
<td>12.0</td>
</tr>
<tr>
<td>Payment*</td>
<td>3</td>
<td>2.7</td>
<td>121</td>
<td>2.2</td>
</tr>
<tr>
<td>Other rewards</td>
<td>1</td>
<td>0.9</td>
<td>24</td>
<td>0.4</td>
</tr>
<tr>
<td>No reward</td>
<td>16</td>
<td>14.4</td>
<td>458</td>
<td>8.3</td>
</tr>
<tr>
<td>Unknown</td>
<td>74</td>
<td>65.5</td>
<td>3493</td>
<td>64.6</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>113</strong></td>
<td><strong>100</strong></td>
<td><strong>5488</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>

*Only students*
Other forms of reward

• Offered the organisation tailored, internal courses as "payment", often used the course exercises as experiments

• Experiments as part of seminars
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)

Simula Experiment Support Environment

Administrator

1: Define experiment
2: Add participants

Researcher

During 3 & 4: Monitor Experiment
5: Collect & analyze results

3:
- Questionnaires
- Task descriptions
- Source code, design documents, etc.

4:
- Answer questions
- Task solutions
- Source code, design documents, etc.
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

Development of SESE

- SESE is built on top of a commercial human resource management system
- SESE is partly being developed by an external company
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.
Professional Project Manager

Simula Scientific Advisory Board, January 2003:
“The recent strategy of paying for professional services, e.g. the implementation of SESE, the analysis of experiments, arranging subjects for experiments, etc. is clearly a good policy and should be continued. However, this does involve a great deal of planning, supervision, interaction and administration. … Taking on public leadership exacerbates these workloads. Dag Sjøberg is in greatest danger from such distracting workloads. It will therefore be worthwhile appointing a project and operations manager, who is sufficiently senior to take the initiative in conducting this work, and to arrange that he or she develops a team that can undertake the planning, external arrangements, conduct, etc. of experiments and can also take a major responsibility for arranging that data is properly curated.”

June 2003:
We employed as Knowledge and Project Manager, Hans C. Benestad, who used to be the Development Manager for one of the largest case tool vendors in Scandinavia. This employment has been very successful.

Example of studies where we have strongly benefited from the full time project manager

- Experiment on pair programming
- DES: study of the development of a web-based database of all the studies conducted at Simula
- SIMBID: study of bidding for the development of a tool to support Simula’s web-based bidding experiments
**DES Study** (Database of Empirical Studies)

- **37**
- Initial tender
- Negotiated contract
- Development
- Acceptance and Deployment
- Operation

March 2003

- Phase 1: Bidding/estimation - full realism experiment

Aug 2003

- Phase 2: UML/Java-based dev. – full realism parallel development multiple-instance case study

Des 2003

- July 2004

- Phase 3: Operation and maintenance
DES phase 1: Full realism experiment on bidding

The bidding process consisted of two separate phases:

In pre-study phase 17 of the 35 bidding companies provided rough non-binding price indications based on a brief, incomplete description of user requirements.

In the bidding phase, all 35 companies provided bids based on a more complete requirement specification that described a software system with substantially more functionality than the system indicated in the pre-study phase.

The 17 companies involved in the pre-study phase presented bids that were on average about 70% higher than the bids of the other companies.

Preliminary theory:
- 1) Software clients tend to achieve better price/uncertainty relationships, i.e., better prices, when the requirement uncertainty perceived by the bidders is low.
- 2) Software clients should not request early price indications based on limited and uncertain information when the final bids can be based on more complete and reliable information.
DES phase 2: Multiple-instance case study with much control

Development process

Characterized by:
• coverage
• density
• emphasis
• sequence

Affects

Project success criteria

• Timeliness
• Costs
• Quality

Rich context

Controlled factors
• requirements
• customer - developer interaction
• team size
• developer skills
• progr. lang (Java)

Uncontrolled factors
• work environment
• company culture
• project organization
• project manager skills
• development tools
• estimate

Moderates how
DES phase 2: Data collection

• About 100 hours of interviews with the four development teams about how they worked.

• Detailed time sheets per individual (split into various development activities and functionalities) were collected daily

• All project documents and code was collected weekly + full CVS history

• Communication between the development teams and the customer was documented using a web-based issue tracking system (400 issues by the end of the projects)

• To evaluate the Java programming skill of the development teams, each developer took part in an experiment in which they were compared with 660 other persons who had previously performed the same task.
SIMBID

• 50 companies invited to give a bid. They were found through a search on the web (“Java” and “outsourcing”) and information on their web-pages. The size of the companies varied from 3-4 to 1000 employees.

• 30 companies accepted: India (13) Ukraine (4), Russia (3), Pakistan (2), Romania (1), Moldovia (1), Phillipines (1), Thailand (1), Poland (1), Belarus (1) and China (1). 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)
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.
Summary: “The Simula opportunity”

- Development of a research group almost from scratch
- Strong research management, focused research
- Strong links to industry, both for research and for technology transfer (experiments, case studies, SPI, seminars, courses, etc.)
- The use of resources mostly up to the department:
  - Only 2/3 of the department’s budget is bound into salaries
  - Extensive use of professionals to take part in experiments and other studies
  - Employment tailored to the needs of the group (e.g., professional project and knowledge manager)
  - Use of consultants for research support
  - Development of sophisticated experiment support environments
- Exploitation of the “brand” “Simula”
Looking forward: What are the main challenges in SE experimentation?

- Developing and testing theories
- Defining and sampling from populations
- Defining the scope of validity of our experiments (necessary to compared and generalise the results)
- Identifying the context (moderator) variables of subjects and objects that may affect the results. (These variables should then be used to characterise populations.)