Experimental Software Engineering: Software Factory Model

Ralph-Johan Back, Ivan Porres and Pekka Abrahamsson

March 16, 2006

Abo Akademi University, Dept. of Computer Science and TUCS
VTT, Oulu
Need for increased research in Software Engineering

- Software is extremely important for the infrastructure of the society

- Software production is a central ingredient in the competitiveness of industry and research today

- Software construction is expensive, time consuming, difficult, and quality of software produced today is low
No facility for experimental Software Engineering

- Need to experiment with different software construction methods and techniques, to improve software production process

- Presently there are no good facilities for experimenting with software processes to find out what works well in practice, and what does not work

- Industry has well defined processes for software production, but researchers cannot use these. Industrial software production is also of rather poor quality, and in need of improvements

- Lack of good experimental facilities leads to a serious gap between theory and practice in Software Engineering research
Software production in academic environment is difficult

- Software is built by students at different levels (undergraduates, M.Sc. students, Ph.D. students),

- Students have almost no prior experience in building larger software systems, i.e., the software is built by novices without a well-defined software construction process

- The result is ad hoc programming, low quality, software that is difficult to maintain after students leave

- This makes it very difficult to carry out long term and ambitious software projects in the academia
Why complain about this now and not earlier

- Software engineering research is industry driven, and large software system failures were felt primarily in the industry.

- The emphasis in academic research on large software system has been on observing software projects, not on building software ourselves in academic environment.

- The academic environment does not have resources to carry out large software projects.

- Building software does not bring academic merits, difficult to publish results in academic journals.
Big science, shared infrastructure

- But other scientific areas are able to carry out large and ambitious research projects

- The large projects are usually centered around centralized and shared infrastructure
  - Astronomy - observatories
  - Nuclear physics - particle accelerators
  - Biochemistry - large biotech labs
  - Medicine - university hospitals
  - Biology - biological stations
A Software Research Initiative

We propose to

- build a *Software Factory* and to
- support *Software Research units* that use the factory.

The primary goal is

- to promote the production of quality software in research and development, and
- to advance software technology in general, both in research and in industry.
Software Factory

• The Software Factory is a production facility, it produces software for different research areas, such as Life sciences, Chemistry, Physics, Computer Science, Engineering, etc.

• It has a well-defined software process and an experienced staff of project managers and software developers, by which it can ensure high quality and maintainability for the software produced.

• The factory gets its assignments from research groups in these areas, and works together with these to build software that they need in their research.
Advantage for R&D

- Professional software support for research will lead to higher quality research software and improve the quality of research itself.

- Investment in research software development will lead to better software production technology and will then benefit the industry at large, through technology transfer.

- Software production is an enabler for new areas of research and for better results in existing research areas.
Software Research Units

- A Software Research Unit uses the software factory as an *experimental facility*, where it can try out new software processes, methods, techniques and tools, and monitor the success of different approaches.

- The units will also carry out larger software development projects themselves.

- Experimentation is done in a careful way, so that the quality of software produced for research groups is maintained: new methods are first tried out in a smaller scale, and once they have proved their worth, they are introduced into standard factory process.
Principal feedback loops
University hospitals

- The closest analogue to the Software Factory is a university hospital.
- The hospital is at the same time a research environment for carrying out controlled experiments, and an ordinary hospital for treating patients.
- This creates a good feedback loop between research and practice.
- Has high volume, to show many different aspects of practice.
- Controlled experiments are carried out, carefully following ethical guidelines.
The Software Factory / University Hospital

- The Software Factory provides the same kind of efficient feedback loop between software research and software practice.

- It would function as a powerful instrument for technology transfer.

- New software construction methods can be first introduced in this environment. Once having shown their merits, they are available for the research community and the industry at large.

- Requires that the Software Factory is realized on a large enough scale.
Business model 1: Software for research purposes

- The Software Factory gets public funding for its basic infrastructure and core personnel.

- Using the public funding the Software Factory can offer its services to research projects at subsidized prices.

- Research projects finance software production in the factory, using their own project funding. This guarantees a steady flow of work assignments and income to the Software Factory.

- Research projects include both basic research projects and industrial research projects.
Overview of business model 1
Comments to business model 1

- Presently there is no supply for this market, industrial software production costs too much for research projects.

- This business model may require changes in funding directives, so that it is ok to pay for software development at the factory.

- To avoid unfair competition with the private software industry, all software that is produced in the factory for subsidized prices could be open source by default. Anybody can benefit from the software that is produced with public funding.

- Closed source software is an option in special circumstances.
Business model 2: Technology transfer

- The Software Factory can also produce software for the industry, but then at full price. The software produced need not be open source.

- The purpose would in this case be technology transfer between industry and the Software Research Center.

- Besides software production, technology transfer could also occur in the form of consultancy, courses and joint research projects.

- Establishment of a national center for software research would benefit the industry at large. Hence, one would expect the industry to also support the software factory financially.
Organization?

- Organizationally a single software factory, to achieve critical mass

- The factory can be geographically distributed over a few places (2-3), depending on the resources allocated to the factory

- The software factory is a central resource for carrying out software experiments, and is shared by a number of different software research units.

- At least one software research laboratory needs to be co-located with the software factory.
Staggered buildup

- Build first up the factory in one location, then expand to one or two other locations, as experience has accumulated (in 2-3 years time)
Producing research software

- Software production for scientific purposes requires close collaboration between software developers and researchers. Need to work out a good software process with strong involvement from researchers. This is partly also a research issue.

- Projects may be in different parts of the country, so the software factory needs to have a distributed form of cooperation with research projects.

- Smaller software projects should be carried out on location, with developers trained at the Software Factory or some of the developers coming directly from the factory. Larger software development projects could be carried out on location at the Software Factory.
How to get there

- A research project is needed to first build up the software factory.

- Could start up with, say, 6 projects:
  - One for building the software factory
  - one for software process development for the factory
  - one for software process measurement in the factory
  - three projects for building specific research applications

- These projects would build up the factory in 2-3 years. Target size for factory would be reached in 4-5 years.

- The factory can gradually be enlarged geographically and with new projects
Alternative rational: Open source software center

- Software factory serves industrial needs
  - a place where industry can build open source software, precompetitive software building
  - TEKES or other industrial funding for longer time

- Software factory serves public needs
  - place where to build open source software for state and municipalities
  - system developed experimentally for one county (hospital, etc.) can be used by others for free
  - possible export of well-fare technology
  - funding by government, municipalities
Educational aspects

- A university hospital also functions as a training center for medical personnel. The software factory could also serve this purpose, as it would employ many CS students.

- The factory would teach best software practices in actual field work, in the context of concrete projects.

- The factory must prioritize the professional quality of its staff, and would therefore only accept the best students, and in limited numbers.

- The factory could serve as a model for software construction, and its method of working and best practices could be taught in courses.
Internal organization of the Software Factory
Funding volume

The financing of the Software Research Initiative can be divided into at least four parts:

- Basic infrastructure funding for the Software Factory (public funding, industry support)

- Research funding for the Software Research Units (public funding, industrial funding)

- Generated income from software production for research projects

- Generated income from consultancy and collaboration with the industry
Permanent personnel for the Software Factory (guestimate)

- 1-2 leaders (director, vice director)
- 4-5 process specialists
- 8 - 10 project managers
- 5-6 senior developers
- 3-4 office personnel
- 3-4 technical support personnel
- 2-3 persons for contacts, public relations, etc.

Alltogether around 30 persons.
Composition of personnel

- The permanent staff is heavy on higher level positions, experts and management.

- In addition, there would be a larger number of personnel employed in software production for research projects and financed by these, typically younger and less experienced personnel.

- Computer Science students would probably be employed in large numbers. They have the basic skills, and would get good training on software development in well-organized environment.
Basic infrastructure funding

- Besides personnel, the center needs office space, equipment, etc.

- A rough estimate is that a fully operational Software Factory would require funding of approximately 2 MEuros/year.

- Assuming that project funded personnel is at least twice as big, this means that 80 - 100 persons would be working at the Software Factory. This on roughly at the same level of personnel as in Frauenhofer IESE (slightly less), although we do not know the financing structure of this institute.

- A suitable initial funding period would be 5-6 years, with a continuation to be negotiated after 3 years, based on initial experiences of the
National Center.
Funding for whole Software Research Initiative

- The Software Factory is the entity which needs separate funding, as an infrastructure for Software Engineering Research, because current research funding instruments (Academy of Finland, TEKES, etc) do not cover it.

- Software Research Units would probably also need additional funding in the Software Research Initiative, to make full use of the new infrastructure, but their activities are better supported by present funding instruments.
Comparison with existing centers

- Software Engineering Institute, Carnegie Mellon University
- NASA Goddard Space Flight Center
- ASU Software factory
- Frauenhofer Institute
- Gaudi Software Factory, Åbo Akademi
- Agile, VTT Oulu
Software Engineering institute

- Software Engineering Institute (SEI) at Carnegie Mellon University. It is a federally funded research and development center sponsored by the U.S. Department of Defense (DoD). The SEI’s core purpose is to help others make measured improvements in their software engineering capabilities.

- The main customer of SEI is the US DoD, while we propose that other scientist and researchers will be the main customers of the Center.

- SEI is focused on software project management, process maturity, and assessment of subcontractors. SEI does not seem to develop software by itself. Our proposal focuses on improving software development methods used to build software as opposed to software project management methods.
The Software Engineering Laboratory at the Goddard Space Flight Center works in close relation with other NASA units where it participates in the development of software.

The laboratory combines a software research facility together with an software development facility. The laboratory is considered a success.

Laboratory has been very successful, but has after 25 years of operation stopped functioning, due to an overall reorganization of NASA and its software production.
Frauenhofer Institute for Experimental Software Engineering

- A German institute that researches advanced topics in software development and software project management.

- It has a unit called he Competence Team Experimentation (EXP) whose objective is to help other research units in the institute to perform empirical experiments.
Finnish centers

- SoberIT, the Software Business and Engineering Laboratory at Helsinki University of Technology has as a goal to improve the global competitiveness of the Finnish software industry by providing education and research.

- AGILE-VTT in Oulu studies how to extend agile solutions to enable their utilization in embedded software development. This project also carries out software case studies in a controlled environment, where software is measured during its development.

- CoSE, the Centre of Software Expertise in Pori is another example, it is concerned with technology transfer in the Satakunta region.
Gaudi Software Factory

- We have built a prototype software factory called Gaudi at Åbo Akademi. This factory is part of the Center of Excellence in Formal Methods in Programming (lead by Ralph Back).

- Gaudi was founded in 2001 and has since then carried out numerous software construction projects, in order to evaluate and provide feedback on new software construction methods. Until now, a total effort of 30 person years has been spent on Gaudi projects.

- The software methods tried in Gaudi have been partly developed at the Center, but most are methods proposed by the Software Engineering research community at large.
Gaudi experience

- This proposal is based on the positive experiences obtained from the Gaudi Software Factory and its synergetic collaboration with the Software Research carried out at the Center of Excellence.
Publications and previous experience


2. Other references to be added as needed.