IESE

Fraunhofer Einrichtung
Experimentelles
Software Engineering

Annual Report 1996

Imprint

Editorial board:
Dr. Frank Bomarius
Dr. Lionel Briand
Dr. Jean-Marc DeBaud
Dipl. Soz. Joachim Müller-Klink

Layout and setting:
Dipl. Soz. Joachim Müller-Klink

Lithography and graphics:
Fotosatz Burkhart GmbH,
Kaiserslautern

Printing:
Printec GmbH, Kaiserslautern

Photography:
Fraunhofer IESE;
Kuhn und Ziller Fotostudio,
Kaiserslautern;
IFA Bilderteam
(cover page: Henseler, Steffl)
University of Kaiserslautern

Annual Report of the
Fraunhofer Institute for
Experimental
Software Engineering
IESE
1996

Fraunhofer Einrichtung für Experimentelles Software Engineering

Sauerwiesen 6
D-67663 Kaiserslautern

phone +49(0)6301/707-100
fax +49(0)6301/707-200

e-mail: info@iese.fhg.de
http: //www.iese.fhg.de

Foreword

In early 1996, the Fraunhofer Institute for Experimental Software Engineering (IESE) was founded in Kaiserslautern. Based on the vision that software competence will become increasingly crucial to the business success of companies in most branches of industry, IESE's mission is to establish itself as a leading organization in applied software engineering research and to become a preferred partner for the transfer of innovative software engineering technologies into industrial practice.

The institute grew out of the successful Software Technology Transfer Initiative at the University of Kaiserslautern (STTI-KL) which was founded in 1993 under the sponsorship of the Ministry of Economic Affairs, Transportation, Agriculture, and Viniculture of the State of Rhineland-Palatinate. Within one year, we achieved significant industrial support from companies in all major branches of industry, established ourselves as a highly-recognized applied research institute, and became an integral part of the information technology infrastructure in our home state of Rhineland-Palatinate.

Experimental Software Engineering employs experiments as an instrument for software technology transfer. Based on the recognition that well-understood and quantitatively manageable software development and maintenance processes need to be customized to a company's specific business goals and characteristics, new and innovative software technologies need to be carefully evaluated before being transferred into practice. After transfer, they need to be continuously optimized based on feedback gained from measurement.

The Fraunhofer IESE provides expertise not only in a wide range of innovative software engineering technologies, but also in approaches concerning the build up of industrial improvement programs for continuous optimization (i.e. TQM, Kaizen) of software development processes. Areas of expertise most sought after by industry in 1996 included software process modeling and measurement as a prerequisite for building up industrial improvement programs. Furthermore, systematic inspection techniques and object-oriented architectural approaches were highly demanded as a means for establishing engineering discipline within software development.

Major achievements in 1996 included the build-up of a highly qualified work force of international standing, the acquisition of industrial projects from companies covering all major branches of industry, and the establishment of an international research reputation. We would like to acknowledge the active support we received from the Central Administration of the Fraunhofer Gesellschaft e.V. in Munich, the University of Kaiserslautern, the State of Rhineland-Palatinate, and our Advisory Board (Kuratorium).

This report is intended to provide you with an overview of our research and transfer work in 1996. Together with the distinguished members of our Advisory Board we will attempt to continue this successful path over the coming years.

Kaiserslautern, August 1997

Prof. Dr. Dieter Rombach

Contents

Vision and Mission

Over the last decades, software has been introduced into almost all high-tech products and services. None of them can function without software anymore. An increasing number of features of these products and services are implemented in software. Consequently, for the majority of industries, for trade, banking, and other service domains, competitiveness and market success depend more and more directly upon their software engineering competence.

Our vision is that software competence will become the most valuable asset of all high-tech product and service branches. Such competence has to be built up, managed, and continuously developed according to well-defined strategic goals. More and more organizations will seek help regarding methods and techniques to identify, customize, continuously optimize and strategically align their software engineering competence.

The mission of the IESE is to establish itself internationally as one of the top addresses both for researchers looking for collaboration in areas of applied software engineering research, and for companies looking for help with their software engineering problems.

The Fraunhofer IESE will be a competent partner in applied software engineering research and technology transfer. In order to live up to this expectation, the IESE has to continuously monitor customers' needs and strategic goals. It has to investigate the most promising innovative software engineering techniques and methods available, to develop their applicability for industrial-strength environments, and, finally, to transfer them into industrial practice, thus building up the software competence sought after by its customers.

Transfer Approach

Since software development is a non-repeatable human-based endeavor, a single standard software engineering technology cannot fit all situations. We strongly believe that high-quality software can only be developed economically by using software engineering technologies tailored to the specific goals and characteristics of the particular development project.

Consequently, software engineering research and transfer need to be performed in an experimental context. Our experimental approach makes it possible to experiment with the technologies in use and thus helps to thoroughly understand their weaknesses and strengths. Technologies can also be tailored to the goals and characteristics of particular projects and organizations and can be packaged together with empirically-gained experience in order to enhance their reuse potential in future projects.

Technology transfer according to the experimental approach follows a three step process.

- New promising technologies and methods are drawn from a rich body of basic research results from the highly-respected Computer Science Department at the University of Kaiserslautern, the special research institute (SFB 501), as well as from research collaborations with many other highly renowned research institutes world-wide.
- Next, the new technologies and methods are experimentally evaluated in laboratory settings, introduced in carefully-selected pilot projects, evaluated in industrial-strength environments, and continuously improved.
- Such validated technologies are then disseminated as best practices to a wider range of customers.

Customer Orientation

Our customers are companies from many different branches, of any size and from a large number of countries. In order to service such a large variety of customers, we have increased our efforts in building up domain knowledge in key application areas such as telecommunications, automotive systems, and banking/insurance/trade, formed a separate service center for small and medium-size companies, and hired scientists from foreign countries to staff international customer projects.

History

The foundations of the experimental approach to software engineering were laid in the Eighties at the Software Engineering Laboratory (SEL), a US organization co-sponsored by NASA´s Goddard Space Flight Center, the Computer Sciences Corporation, and the University of Maryland. The achievements within the SEL were recognized with the 1st IEEE/SEI Process Achievement Award in 1994.

In 1992, Prof. Dr. Dieter Rombach, an active SEL member, moved from the University of Maryland to the University of Kaiserslautern to head the new chair for (Experimental) Software Engineering in the Computer Science Department.

In 1993, he launched the Software Technology Transfer Initiative Kaiserslautern (STTI-KL) which adapted the experimental approach to the needs of German companies and performed numerous successful transfer projects. The STTI-KL was funded by the State Ministry of Economic Affairs, Transportation, Agriculture and Viniculture of Rhineland-Palatinate.

In 1995, the Fraunhofer-Gesellschaft decided to incorporate the successful STTI-KL as a new Fraunhofer Institute. The Fraunhofer Institute for Experimental Software Engineering (IESE) was born.

The IESE is headed by Prof. Dr. Dieter Rombach. In January 1996, the institute started with 14 members. As of December 1996, it employed 33 scientists, 11 non-scientific staff and a large number of students from the University of Kaiserslautern.

Perspective and Agenda

The institute´s strategy is to establish itself as a leading international competence center in software engineering. As of today

- we are coordinator and member of ISERN, the International Software Engineering Research Network, an international network with seventeen members,
- we maintain an international working environment: about one fourth of our staff comes from abroad,
- we are attracting many well-known guest scientists who contribute significantly to the excellence of our institute,
- we are active in many conference committees and editorial boards of international journals.

We will continue and extend these activities.

Concrete next steps in realizing the institute's strategy are:

- Further build-up and continuous improvement of highly-demanded competences in
  - quality and process engineering (quality improvement, quantitative and qualitative methods) and in
  - innovative software product engineering approaches (inspections, product line development, requirement engineering, reengineering and maintenance).
- Concentration on key application domains (telecommunications, automotive systems, banking/insurance/trade).
- Build-up of independent service centers (SME center, training center).
- Foundation of Fraunhofer Centers in the USA and Asia.
- Intensification of research collaborations with international technology transfer and research&development Institutes.

Competence in Research and Technology Transfer

Competence Areas

In order to satisfy the needs of our customers, we have to build up, maintain, and continuously develop a complementary set of competences, namely

- application domain competences
- software engineering competences
- software technology transfer competences.

Application Domain Competences

Our current application domain competence knowledge is concentrated on telecommunications, automotive systems, and banking/insurance/trade.

Software Engineering Competences

The following list provides brief definitions of our key technology competences in Software Quality and Process Engineering:

- Quality Improvement and Experience Factory
  ”Facilitate continuous learning and persistent storage of development know-how.”
- Quantitative and Qualitative Analysis
  ”Capture relevant development data and analyze them.”
- Process Modeling
  ”Represent key business and software development processes.”
- Integrated Software Engineering Environments
  ”Support all of the above.”

Key technology competences in Innovative Product Engineering are briefly defined as follows:

- Requirements Engineering
  ”Improve the early phases of software development.”
- Product Line Approaches
  ”Structure domain and design knowledge as well as software development know-how in such a way that it can be easily understood, changed, and reused across families of systems.”
- Reengineering and Maintenance
  ”Redocument and transition legacy systems and manage long-living software systems.”
- Cleanroom Software Engineering
  ”Develop certifiable and reliable software.”

Technology Transfer Competences

Transfer of advanced industrial-strength software engineering technologies is the central task of the Fraunhofer IESE. We therefore maintain a transfer-oriented network of collaborations with technology providers, such as universities, with research and development departments of large organizations, with providers of tools that support our technologies, and with strategic partners that otherwise support our work.

Competence gained from collaboration with these providers enables the IESE to conduct technology transfer projects with customers, i.e., the users of our technology.

On the technology side, we have to monitor the latest developments, identify promising technologies, and experimentally evaluate and improve them to create industrial-strength technologies.

On the customer side, our competences are to identify strengths and weaknesses of organizations, to define strategic improvement goals with our customers, to implement continuous improvement programs, to set up means to monitor progress of the changes introduced, and to capture and store experiences made.

Collaborations

The IESE conducts collaborations with technology providers, technology-transfer customers, and strategic partners. The overall goal is to identify, further develop, and put into industrial prac-tice software engineering technology so as to increase the competence of our customers.

International Research

Among the international cooperations in applied software engineering research, the International Software Engineering Research Network (ISERN) with about 20 sites in research and industry plays a prominent role. ISERN is a forum for applied software engineering research with members from Europe, America, Asia, and Australia. It maintains high-level contacts to leading international companies in the embedded systems domain such as AT&T, Motorola, Nokia, Ericsson, NTT, Matsushita, Hitachi, and Daimler-Benz.

Publicly-funded Collaborations

Collaborations exist with many publicly-funded consortia aimed at either software engineering technology advancement or dissemination of best practices. Publicly-funded projects can be devoted to research and development as well as technology transfer. Often, additional bilateral industrially-funded collaborations result from performing these projects. Public project sponsors include the Government of the State of Rhineland-Palatinate, the Federal Government of Germany, and the European Commission.

Industrially-funded Collaborations

In the first year of its existence, the institute already developed 14 bilateral industrial collaborations with leading companies offering software-intensive products, software-intensive services or software products. In addition, collaborations exist with 12 industrial companies which are partially supported through publicly-funded projects.

The overall IESE approach is well appreciated by all its customers. This is documented mainly by their renewed business (i.e., prolongation of contracts) and a strong increase in the number and volume of industrially-funded projects. The cooperation partners of the Fraunhofer IESE range from very large global players to very small companies. They can be roughly grouped into four categories:

- Large national and international companies that seek help in their mid- to long-term endeavor of quality improvement in software development.
- Large national and international companies that can afford their own R & D departments and that search for competent research partners.
- Medium-size companies that want to set up improvement programs but are usually under very tight budget and schedule constraints.
- Small companies that need ready-to-use, evaluated technologies which yield short-term return on investment.

Offerings

For developers of software, we offer:

- the evaluation of software development practices,
- the construction of customized quality improvement systems,
- the introduction and optimization of engineering-based, state-of-the-art software development processes,
- support towards development of certifiable software,
- preparation for certification.

For users of software, we offer:

- help in purchasing software,
- independent support for monitoring software development contracts.

For small and medium-size enterprises (SMEs) we offer individual assistance and "products" tailored specifically to their needs on request.

Our services are offered by means of:

- goal-oriented transfer projects,
- long-term strategic research and development alliances,
- consulting,
- executive briefings,
- continuous training and education,
- studies and expert reports,
- state-of-the-art surveys,
- product evaluation,
- prototypical tools.

A Matrix Organization

The structure of the Fraunhofer IESE is designed to optimally support applied research and technology transfer projects. In order to serve the differing needs of our customers, we have to put together project groups in a very flexible way. If necessary, we reorganize the groups to accommodate the needs of the projects that change over time. Therefore, the basic structure is a matrix organization. Through the matrix we bring together our software engineering competences on one side - provided by the Quality and Process Engineering (QPE) and Innovative Software Engineering Approaches (ISE) departments - and the application domain know-how on the other side - provided by the Industrial Quality Improvement Projects (IQVP) department.

While QPE and ISE are maintaining and continuously improving the institute's technical competences, the department for Industrial Quality Improvement Projects IQVP (Industrielle Qualitäts-Verbesserungs-Projekte) and our fourth department, Central Services and Public Projects ZDÖP (Zentrale Dienste und Öffentliche Projekte) are responsible for successfully conducting industrially- and publicly-funded projects.

The IQVP department is the competence center for application domain know-how. ZDÖP adds competence regarding the acquisition and management of publicly-funded projects. Both ZDÖP and IQVP provide man power for managing research and transfer projects.

Research and technology transfer is supported by administrative and technical groups and centers:

- Public Relations is concerned with media presence, the preparation of printed material about the institute, and the organization of exhibitions.
- The Demonstration Center provides on-line sample implementations of tools, data, and other tangible results of our research and project work.
- The Training and Education Center improves the training and consulting competence of all members of the institute. It develops the institute's human resources by organizing training and seminars, and it assists in creating professional training materials for our customers.
- The Consulting Center for Small and Medium-Size Enterprises (SMEs) is specifically designed to take care of the particular needs of small and medium-size companies. Industrial transfer projects are conducted here, like in IQVP, but for SMEs.

The European Public Projects and National Public Projects groups within ZDÖP are concerned with applications for public funds and with the special terms of conducting such projects.

The groups within IQVP address the special needs of different customer groups. Customers that develop software as an add-on secondary product can be grouped into those who build systems that strongly rely on software and those who provide software dependent services. Due to the relativly large number of customers from the telecom domain, we have a particular group to support them. The Software Providers Group handles projects with companies that develop software as their primary product.

Organigram

Fraunhofer Institute for Experimental Software Engineering IESE
Prof. Dr. D. Rombach

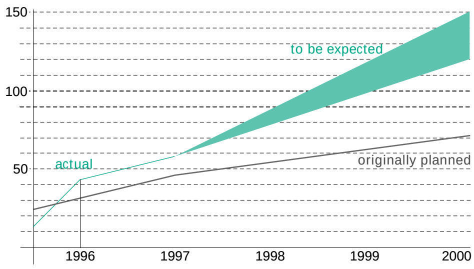
- **External Relations Department** (EB) – Prof. Dr. Rombach
  - Acquisition – Prof. Dr. Rombach
  - Public Relations – Müller-Klink
  - Demonstration Center – Dr. Rösch
  - Education & Training Center – Eberle
  - Consulting Center for SMEs – Prof. Dr. Rombach
  - Contact Office University of Kaiserslautern – Prof. Dr. Rombach
- **Central Services & Public Projects Department** (ZDÖP) – Dr. Ruhe
  - Administration – Halle
  - Library & Publication Service – Dr. Ruhe
  - Technical Service – Huber
  - European Public Projects – Dr. Ruhe
  - National Public Projects – Dr. Ruhe
- **Quality & Process Engineering Department** (QPE) – Dr. Briand
  - Quality Improvement & Experience Factory – Dr. Briand
  - Quantitative Methods – Dr. El Emam
  - Process Modeling – Dr. Briand
  - Software Engineering Environment – Dr. Rösch
- **Innovative Software Engineering Approaches Dept.** (ISE) – Dr. DeBaud
  - Cleanroom – Prof. Dr. Rombach
  - Requirements Engineering – Dr. DeBaud
  - Product Line Approaches – Dr. DeBaud
  - Reengineering & Maintenance – Dr. DeBaud
- **Industrial Quality Improvement Projects Dept.** (IQVP) – Dr. Bomarius
  - Software-Intensive Products – Dr. Bomarius
  - Software-Intensive Services – Dr. Bomarius
  - Telecommunication Services – Dr. Schwarz
  - Software Providers – Dr. Bomarius

Personnel

As a result of the number of acquired projects, the Fraunhofer IESE has almost doubled its staff during its first year. Our demanding requirements forced us to hire leading experts from many different nationalities, thus creating an extraordinary multinational culture at the institute.

Based on the very encouraging acceptance of the Fraunhofer IESE by industries, we predict a steady growth of the institute to a staff of approximately 120 in the year 2000.

Staff	Scientists	Infrastructure	Guest Scientists	Students and employees	Sum
01.01.1996	11	2	0	1	14
31.12.1996	30	11	3	23	67
Plan 1997	38	15	5	30	88
Plan 1998	44	17	7	38	106



Budget

Income	TDM	%
Industrially-funded projects	1 443	37.6
Publicly-funded projects	403	10.5
Other Income	382	9.9
Public Grant (State of Rhineland-Palatinate)	1 612	42.0
	3 840	**100.0**

Expenses	TDM	%
Personnel	2 700	70.3
Miscellaneous	1 140	29.7
	3 840	**100**

Investments

Income	TDM	%
Industrially-& Publicly-funded projects	82	2.9
Public Grant (State of Rhineland-Palatinate)	2 698	97.1
	2 780	**100.0**

Expenses	TDM	%
	2 780	**100**

Advisory Board

Research

Prof. Dr. Victor Basili
Institute for Advanced Computer Science
Department of Computer Science
University of Maryland
USA

Prof. Dr. Jürgen Nehmer
Department of Computer Science,
University of Kaiserslautern,
also: Member of the German Science Council (Deutscher Wissenschaftsrat)

Vice-Chairman of the Advisory Board

Industry

Prof. Dr. Ernst Denert
Chairman, sd&m GmbH & Co. KG
software design & management
also: Vice-President of the German Computer Society "GI" (Gesellschaft für Informatik)

Chairman of the Advisory Board

Dietmar Freigang
IS Director
Allianz-Lebensversicherung AG

Günther Plapp
Technical Director, K3/LE
Robert Bosch GmbH

Prof. Dr. Eckart Raubold
Director, Technology Center Darmstadt
Deutsche Telekom AG

Government

Brigitte Klempt
Representative of the Ministry of Education, Science and Continuous Education of the State of Rhineland-Palatinate

Dr. Ulrich Müller
Representative of the Ministry of Economic Affairs, Transportation, Agriculture and Viniculture of the State of Rhineland-Palatinate

Technical Competence Areas

Research Mission

Our customers face real, large-scale quality, productivity and time-to-market problems within their software divisions. They expect us to perform quick root cause analyses, propose adequate techniques, methods, and tools to mitigate the identified problems, and help integrate them into their software and business processes as manageable competences.

This implies that such technologies should be rigorously evaluated within realistic conditions and properly packaged. In addition, once transferred, these technologies must be tightly controlled and managed for optimal use. That is, we must ensure that these technologies are properly used with respect to: conformance to intended use, resource expenditures, organization issues, and quality objectives.

The core technical contribution of the IESE is to empirically characterize, validate, and package innovative software technologies. To address this goal, the IESE core technical competences are structured around two highly synergistic departments: Quality and Process Engineering (QPE) and Innovative Software Engineering (ISE).

The primary mission of Quality and Process Engineering is to provide support for the transfer of software technologies through rigorous evaluation, quality control, and quantitative management.

The primary mission of Innovative Software Engineering is to develop a portfolio of effective and innovative software engineering methods and techniques for careful evaluation and transfer purposes.

Quality and Process Engineering

In order to monitor, evaluate, and control the transfer and tailoring of software technologies (e.g., tools, processes) into an organization, one needs to be able to measure the strengths and benefits of such a technology, but also its costs and inherent risks. In addition, in order to determine how to integrate a new technology into current practice, one needs to understand the software development processes and technologies in place, understand their weaknesses and strengths. Thus, the potential gains and dangers of a new technology can, in context, be precisely assessed and quantitatively investigated.

Once transferred, any technology needs to be monitored, controlled, and managed in order for it to be effective from a quality and productivity standpoint. In order to do so, the resource consumption and quality achievements of a technology need to be quantitatively modeled and linked to contextual and human factors.

To address the issues mentioned above, the QPE department is composed of four interacting and complementary groups:

- Quantitative Methods (QM)
  QM focuses on ways to build quantitative models aimed at the monitoring, evaluation, and prediction of software quality and productivity attributes such as productivity, maintainability, reliability, and related software risks. This implies the use of measurement, statistical modeling, and many other experimental techniques.

- Process Modeling (PM)
  PM aims at providing methods for process elicitation, modeling, and analysis so that specific process weaknesses and strengths may be identified. This is expected to naturally drive process improvement initiatives. We put particular emphasis on developing techniques to cope with real-scale, high-complexity processes and organizations.

- Quality Improvement and Experience Factory (QE)
  QE aims at supporting the packaging of software knowledge (e.g., process models, productivity models) within an organization and facilitate its reuse. The dissemination and proper reuse of software knowledge requires the provision of facilities to store all forms of knowledge, retrieve and tailor it. In addition, QE provides strategies, infrastructures, and methods to support long-term cumulative, organizational learning.

- Software Engineering Environments (SEE)
  SEE aims at providing automated support for all the activities described above, ensuring that they can be implemented at minimal cost.

We place particular emphasis on developing solutions that are technically sound, optimal in their specific context of application, and tailored to our customers' expectations and needs.

Dr. Lionel Briand, Department Head

Quality Improvement and Experience Factory Group

One of the fundamental premises of experimental software engineering is that we wish to understand and improve software quality and productivity. This must be based on empirical evidence and project experience. Even for small software organizations, large amounts of information can be built up over the years (e.g., expertise, project data, lessons learned, quality models). For such information to be usable, it needs to be modeled, structured, generalized, and stored in a reusable form in order to allow the effective retrieval of relevant artifacts. A continuous build-up of knowledge requires an approximate organizational structure which must be integrated with the software development organization. Such an organizational structure dedicated to the acquisition and reuse of experience together with its underlying experience base technology is referred to as an experience factory.

Goal

We define an efficient organizational strategy and structure. This is the operational backbone of the experience factory in a software organization. In addition, we are developing new technologies to capture, model, store, generalize, retrieve, integrate, and update software engineering knowledge. These technologies must be combined within dedicated experience bases and tailored to the needs of the respective software organizations.

Description

This group focuses on the following issues:

- Designing an organizational strategy and structure of an experience factory in such a way that it can be tailored to the specific needs of the respective software organization.

- Designing software engineering experience bases with adequate knowledge representation and efficient knowledge base technology.

- Identifying strategies for feeding and updating a software engineering experience base as well as retrieving information from it. Such strategies should be suitable for various application contexts, e.g., the planning and monitoring of projects, the à posteriori analysis of projects, and the à priori definition of measurement programs and process improvement plans for new projects.

Scientific Issues

- Which activities, roles, and competences are necessary to operationalize an experience factory, especially, what are their relationships, which products are accessed, and what are the respective information flows? Given the specific requirements and preferences of a software organization, to which degree can the respective activities be automatically supported?

- What should be the structure of a software engineering experience base, e.g., what information and experiences should be stored, how should they be captured and modeled? For instance, how should experience regarding a particular development technology (e.g., inspection, design) be modeled and made reusable?

- How should the representation, update, and retrieval of relevant information be supported to deal with the cognitive complexity of a large and diverse corporate information base? For instance, when planning a measurement program, what artifacts (models, measures, etc.) can be reused from previous measurement experiences?

- How can different kinds of experiences (i.e. the objects to be stored in the experience base) as well as different kinds of representations of specific experiences (e.g., a measurement plan described on different levels of abstraction; more or less vague knowledge collected as lessons learned during the introduction of a new software engineering technology) be made available and combined in an effective way that also considers the respective context of the actual user?

- Which kinds of candidate technologies for supporting the implementation of an experience base do exist and how do they relate to one another? What are their respective costs and benefits? How do they correspond to the specific requirements of an industrial software organization?

Practical Use

Such software engineering knowledge bases can be used

- to support project management decisions, e.g., which technology to use on a project, and

- to identify and drive structural and technological changes in a software organization, e.g., identify high payoff areas for improvement.

As soon as the necessary knowledge has been captured, identified, formalized, and represented (e.g., as products, models, lessons learned), an experience base can be constructed. Thus, the knowledge acquired in such programs can be made a corporate asset and becomes widely available across the organization.

Cooperation

Research Cooperation:

– Software Engineering Laboratory (SEL), University of Maryland, College Park, Maryland (USA);
– Center for Learning Systems and Applications (LSA), University of Kaiserslautern, Kaiserslautern (D);
– Federal University of Santa Catarina, Florianopolis, (Brazil)

Andreas Birk

Christiane Gresse

Carsten Tautz

Quantitative Methods Group

For software organizations to improve their efficiency and effectiveness, they have to be able to measure their processes and products. This measurement can be used to characterize and baseline their processes and products, to evaluate new technologies, to identify and track improvements, and to monitor and control projects. This approach is referred to as measurement-based improvement.

Goal

The general goal of the Quantitative Methods Group is to develop technologies that facilitate measurement-based process and product improvement.

Description

This group is involved in the following technology transfer and research areas:

- setting up measurement programs in industry following the GQM paradigm and generalizing from these experiences to identify critical success factors for setting up a measurement program,

- empirical evaluation of software products through the analysis of field data and through field and controlled experiments,

- developing techniques for the measurement, evaluation, and control of software inspections through the analysis of field data and through field and controlled experiments,

- developing and evaluating modeling techniques for software cost estimation,

- developing more modeling techniques for other topics.

Scientific Issues

The Quantitative Methods Group identified the following areas as subjects of future research activities:

- Definition and Validation of Product Measures
  "Which measures are most useful for understanding the structure of software and for managing its quality?"

- Inspections
  "What criteria should be used to decide whether to reinspect a software artifact?" and "How do we assess the effectiveness and efficiency of software inspections?"

- Cost Estimation Models
  "Which modeling techniques provide the greatest cost estimation accuracy?" and "How can we incorporate local expert knowledge effectively into cost estimation models?"

- Data Analysis
  "Which machine learning and statistical data analysis techniques are most suitable for solving particular problems?" and "How can we improve upon existing data analysis techniques for use with software engineering data?"

Practical Uses

- Product evaluation
  We have built models to predict the error proneness of software components from measures of the system design, and we have compared object-oriented design documents built using different design

guidelines to see which guidelines yield less maintenance effort.

- Measurement, evaluation, and control of software inspections
  We have built models to estimate how many defects are remaining in a document after an inspection to help decide whether it ought to be reinspected, and we have developed a benchmarking approach whereby the performance of inspections can be compared to that of companies that already have implemented inspections successfully.

- Cost estimation
  For one organization that had a small historical data set, we developed a cost estimation model that augmented this data with the knowledge of experienced project managers, and in another project we developed an effort estimation model using functional size measures by combining machine learning techniques and statistical techniques.

Cooperation

Research Cooperation:
- European Software Institute, Bilbao (E);
- Software Technology Transfer Finland, Espoo (SF);
- GrafP Technologies Inc., Montreal, Quebec (CAN),
- Centre de Recherche Informatique de Montreal, Montreal, Quebec (CAN);
- Software Engineering Institute (SEI), Carnegie Mellon University, Pittsburgh, Pennsylvania (USA);
- Software Engineering Laboratory (SEL), University of Maryland, College Park, Maryland (USA)

Industrial Cooperation:
- Daimler-Benz AG, Forschung und Technik, Ulm (D);
- Daimler-Benz Aerospace DASA, Bremen (D);
- Robert Bosch GmbH, Frankfurt/Main (D);
- sd&m GmbH & Co. KG, Munich (D)

Dr. Khaled El Emam, Group Leader

Dr. John Daly

Pierfrancesco Fusaro

Isabella Wieczorek

Software Engineering Environment Group

Software engineering involves many different complex activities and artifacts concerning both product (e.g. documents, code) and process (e.g. design and inspection activities and their relationships) engineering.
To keep the software engineering process under control and efficient, information about activities and products must be entered, stored, and maintained in computer-based environments. Software engineering environments help engineers deal with the inherent complexity of software engineering by providing computer-based tools for all areas of software engineering in an integrated environment. This includes tools for all aspects of product and process engineering.

Goals

The software engineering environments group addresses tool-related aspects that are relevant in the context of the software development process. The following specific domains are target goals:

- Computer-Aided Software Process Elicitation:
  Support improvement programs by understandable, trackable and easily modifiable representations of the software process.

- Computer-Aided Software Process Measurement and Measurement Planning:
  Support integration of process modeling, measurement and goal-oriented measurement planning.

- Integrated Software Process Management Support:
  Support the integrated management of software process models, measurement plans, measurement data, and analysis results.

- Evaluation of Software Engineering Environments:
  Analyze existing software engineering environments, technologies and architectures, focused on software development process and process improvement aspects.

- Prototyping of Software Engineering Environments:
  Build prototypes of tools or environments to gain more insight in promising concepts and to help support and improve the work in industrially/publicly-funded projects.

Description

The QPE/SEE group tries to build up competence in the area of software engineering environments. Therefore the group builds concrete prototypes of environments and maintains knowledge about Software Engineering Environments in general. The group also acts as central coordinator with respect to tool development in the institute.

The QPE/SEE is involved in the following specific activities:

- Spearmint
  Prototype supporting Software Process Elicitation, Analysis, Reuse, and Measurement in an integrated environment.
  In the Spearmint project, a prototype for comprehensive software process support is being developed. The need for such a new product was caused by the lack of support given by existing tools for elicitation and analysis of software processes. The focus of the Spearmint project is to support

software process elicitation, measurement planning, analysis and reuse. The Spearmint prototype provides multiple configurable views on the software process in an integrated and comprehensive environment. The environment includes features like database and querying support, change tracking, and hypermedia support.

- Demo-Lab/Tool Suite
  Tool evaluation and demonstration platform.
  Web-based environment that lets users easily access tools, get information about them (i.e. description, experience reports), and guides the user through a demonstration session, giving a quick and practical overview about current state-of-the-practice in this field.

- Demo-Lab/Hyperteaching Environment
  Multi-media environment for demonstration of experimental software engineering principles and of IESE's technology approach.

Scientific Issues

- Computer-Aided Software Process Elicitation
  How can information about a software process (e.g. models, plans, analysis results) be represented in an understandable way? How can we support the interaction between humans and a software process support tool efficiently?

- Computer-Aided Software Process Measurement and Measurement Planning
  How to integrate measurement into process models and how to navigate effectively process models, measurement plans, and measured data?

- Integrated Software Process Management Support
  Which are the right techniques for supporting process model reuse? How can we integrate facilities for information tracking, navigation, querying, and versioning into a comprehensive environment?

- Evaluation of SEEs
  What are reliable/objective evaluation criteria?

- Prototyping
  Which technologies are appropriate for rapid prototyping?

Practical Use

Process improvement can only be efficiently achieved in practice if it is well supported by tools. Software engineering environments provide the basis for gaining intellectual control over processes and products.

The SEE group provides some of these tools through the Spearmint project. The Spearmint prototype is targeted to be a product which can be used by our scientists in actual industrially and publicly-funded projects.

Besides the practical use in projects, the Spearmint project also provides a defined architecture and integrated framework for the building of software engineering environments. Therefore, Spearmint represents an integration framework for tools which are developed at our institute.

Cooperation

- University of Kaiserslautern, Kaiserslautern (D);
- University of New South Wales, Sydney (AU)

Dr. Peter Rösch, Group Leader

Ralf Kempkens

Dr. Richard Webby

Jörg Zettel

Process Modeling & Analysis Group

Software development organizations operate according to a set of complex processes like specification, design, coding, testing, and configuration control. The introduction of new technologies or methods in this environment often causes severe problems. One of the major reasons is the lack of explicit information about the actual processes and their interrelationships.

Visualization of processes is needed first of all to understand and then to better control the change of software development organizations and to predict the performance of projects. This visualization is achieved by modeling the processes descriptively. Simulation of those models allows the observation of their behavior in different contexts at low cost. Analysis of the models uncovers problems in software development.

Process modeling provides a necessary prerequisite to better understand, control, manage, and change software development processes. Overall, the explicit process models represent an important part of an organization's experience and demonstrate its ability to handle complex tasks. Process models are therefore a basic part of the documentation which is required for ISO 9000 certification. Availability of such documentation and fidelity to the processes executed are an important quality aspect in the assessment of an organization.

Goals

The general goal of the Process Modeling and Analysis Group is to provide accurate models of software development processes for use in process improvement. This consists of the following elements:

- Modeling
  Making software industry processes explicit, defining and validating conceptual frameworks for process modeling, providing a set of knowledge acquisition techniques for elicitation of process information, defining a process modeling methodology, and evaluating tools for process modeling.

- Analysis
  Analyze the process models both quantitatively and qualitatively in order to identify where process improvement is necessary, compare candidate process models for selection of best practice development.

- Process management
  Innovative technology to store, retrieve, and tailor process knowledge.

Description

The Process Modeling Group is involved in the following technology transfer and research areas:

- Modeling industrial development processes, including the evaluation of applied process modeling technology.

- Development of a conceptual schema that helps capture all relevant aspects of processes within improvement programs, which includes the transformation between different process modeling languages.

- Specifying, testing, and evaluating process modeling tools for descriptive process modeling.

- Examination of mechanisms for coping with process variants and versions, which includes project-specific tailoring of process models.

- Definition of an operational process modeling method.

- Development of a WWW-based process guide, which is a structure and navigation mechanism to represent an organization's process models and to support developers in their daily work and which helps them achieve process conformance.

Scientific Issues

The Process Modeling and Analysis Group identified the following research fields as subjects of their future research activities.

- Conceptual Model and Views
  "Is there a canonical schema for describing development processes and what are useful variants in different contexts?"

- Modeling Method
  "What techniques are candidates for extending the portfolio of technologies for process elicitation and process documentation?"

- Analysis
  "What project parameters influence product quality and process attributes (e.g., timeliness, effort) and how can they be captured in process models?"

- Process Management
  "What are the requirements for defining an instance of an experience factory for process models?"

Practical Uses

Process modeling helps companies to understand the complex relationships in software development. Explicit process models support communication among different roles (e.g., managers and developers) and help reconcile differing views on the software process. They help project planning, identify causes for low product quality or budget overrun, capture the experience of an organization, fulfill requirements for quality management (e.g., ISO 9000-1), and support the implementation of process change within improvement programs.

Cooperation:

Research Cooperation:

– Daimler-Benz Forschung und Technik, Ulm (D)
– Software Engineering Institute (SEI), Carnegie Mellon University, Pittsburgh, Pennsylvania (USA);
– University of Kaiserslautern, Kaiserslautern (D)

Industrial Cooperation:

– Daimler-Benz Aerospace DASA, Bremen (D);
– Robert Bosch GmbH, Frankfurt/Main (D)

Ulrike Becker

Dirk Hamann

Dietmar Pfahl

Building of the Fraunhofer Institute for Experimental Software Engineering in Kaiserslautern

Location of the Contact Office of the Fraunhofer IESE at the University of Kaiserslautern

Innovative Software Engineering

The primary mission of the IESE Department for Innovative Software Engineering (ISE) is to develop a portfolio of effective and innovative software engineering methods and techniques for careful evaluation and transfer purposes. To achieve this mission, the ISE uses its customers' existing or anticipated needs as the principal driver when monitoring the state of the art, when selecting, adapting, and packaging promising approaches or, when required, developing new ones.

In order to structure the department, our strategy has been to adopt an extended life-cycle coverage view. This view enables the construction of a rich and wide-enough portfolio of approaches that can address a large variety of the problems faced by our customers as well as anticipating emerging needs in our field of endeavor. To reflect this strategy, we have organized the department into four complementary groups:

- The Requirements Engineering group (RE)
  focuses on the extraction and specification of what precisely a system should do.

- The Cleanroom group (CR)
  addresses questions related to the systematic, traceable development of requirements into a system, with certifiable attributes whenever possible. The CR group also tackles software inspections. Inspections are aimed at the elimination of errors and can basically be performed at any time within the lifecycle.

- The Product Line Approaches group (PLA)
  helps organizations efficiently produce systems when these share many similarities. These reuse approaches encompass domain engineering as well as domain-specific software architecture issues.

- The Software Maintenance and Reengineering group (RM)
  tackles the multiple types of problems that arise when a system has been fielded and needs to be maintained, redocumented, and/or restructured.

The formation of the ISE department started later than the other IESE departments, hence by the end of 1996 the ISE is still in the process of developing its portfolio of approaches, increasing staff and fostering its external research relationships.

We put strong emphasis on creating tight relationships with our customers. We fully appreciate both the difficulties but especially the opportunities of such close collaborations, and we are developing an internal culture to capitalize on them. Finally, we have learned that being practical and problem-oriented is the key to our success.

Dr. Jean-Marc DeBaud, Department Head

Product Line Approaches Group

There is tremendous customer pressure on the market for organizations to develop multiple variants of their core products, which should, of course, be ever faster, cheaper and of higher quality. Historically, organizations have responded to this challenge by organizing their products (and hence competencies) around product lines. There, variants are derived and manufactured from a core structure, or architecture.

This pressure also affects the development of software systems and, one may argue, is even more prevalent here than in most other industries. The vision of product line approaches for software systems (PL) is to enable organizations to manage their software development efforts according to and benefiting from the PL principles. PL's principal goal is to manage product variability while minimizing effort duplication and maintaining an open and flexible central design.

Reuse is central to the idea of achieving control over a PL. Successful reuse takes many forms, but it has become apparent that analysis and design reuse, beyond the more traditional reuse of code, hold the key to achieving systematic and widespread software reuse.

In order to reuse analysis and design, it is necessary to take a domain (the business or scientific application area of the product line) view of the world. A domain is an abstraction that denotes a set of similar problems that are together deemed to share a number of fundamental characteristics. Examples of such domain are Avionics, Accounting, Warehousing, and Guidance Systems.

Problems within a domain have often been solved over and over again. Hence, it may be possible, via a domain engineering process, to extract the specification and structure of one or more generic solution designs covering the problems in the domain. This almost always takes the form of a generic (reference) architecture specification. This reference architecture then serves as the PL core (reuse infrastructure) with which subsequent system variants can be efficiently derived. The field of PL approaches encompasses all the steps and processes necessary to construct and use such a domain-based reuse repository.

Goals

The main goals of this group are:

- Technical Issues
  Understand better the practical processes used and the problems encountered behind constructing, using, and evolving over time PL for software systems.

- Technology transfer issues
  Acquire, analyze, and reuse process information relating to the development of PL within a variety of organizations.

Both goals provide the enabling conditions for the successful transition of this type of domain-based intensive reuse within organizations. We use the original DARPA-DSSA program as a foundation of our work in this area.

Description

In particular, this group focuses on:

- The development of a multi-step process for introducing PL within organizations.

- Identifying the type of information that must be recorded when performing commonality analyzes as well as the most-suited representation notation.

- Which of the existing domain analysis methods appears to work best in practice, both from a scientific and a practical point of view? What types of information must be captured and how? Does this depend on the particular domain?

- How to transition results of a domain analysis to a fully-documented robust, flexible and highly customizable reference architecture.

- Understanding how variable designs can be expressed and understanding the usability aspect of the notation used. Is object-orientation suitable for this task?

- Maintaining traceability from the reference requirements down to the executable components of the reference architecture.

- Managing the versioning problem created by using a reuse infrastructure to develop a family of systems when the infrastructure itself evolves rapidly.

- How can product line designs be evaluated?

Scientific Issues

- How can we provide a scientific foundation for the notion of domain? This includes scoping, modeling, and representation issues that are difficult to resolve fully: There is some evidence to suggest that different types of domains call for different answers. What are the key issues characterizing domain types, and what consequences does this have?

- The level of complexity needed to drive the evolution of a PL is dependent on the maturity of the domain and the quality of the domain engineering leading to its definition. When domain maturity is low or modeling was less than optimal, perhaps for economic reasons, deep changes are necessary to the reference architecture. We need to find a way to tackle this systematically.

Practical Use

PL for software systems is an architecture-centric, reuse-based software development approach. The focus is on capturing and exploiting an organization's core business competence areas. To an organization subscribing to the approach, PL provides a method to analyze and record the organization's competence as well as a process for constructing a software reuse environment that will be key to the organization's software development capabilities in that domain and hence to its competitiveness.

Cooperation

Research Cooperation:

- Software Engineering Institute (SEI), Carnegie Mellon University, Pittsburgh, Pennsylvania (USA);
- Semantics Designs, Austin, Texas (USA);
- University of Kaiserslautern, Kaiserslautern (D)

Industrial Cooperation:

- Ericsson Eurolab Deutschland GmbH, Herzogenrath (D);
- Markant Südwest Software- und Dienstleistungs GmbH, Kaiserslautern (D);
- Tecmath GmbH, Kaiserslautern (D);
- Variant-building Project: Six Local SMEs.

Oliver Flege

Reengineering and Maintenance Group

A large part of today's programmer time is spent maintaining legacy systems. It is estimated that 40 to 70 % of software development organization costs are spent on maintenance.

Indeed, successful software systems have a long lifetime of modifications. These are often essential to companies' business. However, unless special care is taken, the quality of a software system (modularity, cohesion of its components, understandability, ...) decreases and the cost of maintenance increases dramatically.

Two strategies can help correct this situation:

- Better maintenance practice: Establishing practices to reduce system degradation.

- Redocumentation:
  A major portion of the large maintenance effort is spent on understanding the existing program and the data manipulated.

Within this context, helping maintainers preserve the quality of legacy systems can be achieved by helping them to better understand the legacy systems and thus enable them to predict how proposed changes to services would impact the system and could lead to side effects.

Goals

The main goals of this group are:

- Better understand what constitutes a good practical system overview for a maintainer who has to understand and modify a system under time pressure. In particular, should this overview contain the structure and dynamic behavior of the system, contracts between its components, data view, etc. and in what form.

- Understand what type of analysis can suggest transformation to a system which would improve its maintainability and would preserve other quality features.

- Study a number of specific approaches and techniques to preserve the quality of a software system during maintenance:
  - change management
  - change request tracking systems
  - measurements related to changes
  - maintenance processes models
  - regression testing
  - version and configuration management
  - scenario-based input analysis.

Each goal has a particular industrial perspective. It focuses on acquiring, analyzing, and reusing process information on the introduction of these approaches and techniques in an industrial setting.

Group Focus

In particular, this group focuses on:

- Extracting architectural views of a system
  - What are the components of this system?
  - How do they relate?
  - What are the contracts (protocols) which constrain the interaction among components?

- Integrating and maintaining user's input on a partially-recovered architecture. This user input is crucial, because semantic information can be gained which plays a role in the organization of an architecture and cannot effectively be recovered otherwise.

- Integration of domain information in architectural recovery tools.

- Extraction of data dependencies and classification of system's variables according to their role in the system as a complementary system view.

- Identification of code which has been duplicated, then slightly modified. This code often leads to maintenance problems, since when a bug is corrected the maintainer does not know how often this bug has been duplicated and where else it needs to be corrected.

- Creation of prediction models for maintenance efforts and programming guidelines derived from the predicted efforts.

- Analysis of the structure of a system to suggest ways to perform information hiding and separation of concerns.

Scientific Issues

- Which of the existing architectural description languages expresses best the structure, behavior and constraints of an architecture, so that it can be used and accepted in industry?

- How can a domain model guide architectural recovery and how can architectural recovery tools contribute to the creation of such a model?

Practical Use

Provide an architectural description of a system to support

- validation of change (did we break or degrade the architecture?),
- analysis of scenarios of possible future changes,
- testing,
- creation of a model for a family of systems in the same application domain,
- identification of cloned code in a system to reduce the size of a system a maintainer has to oversee and reduce the number of places where duplicated errors have to be corrected,
- identification of reusable components from existing system along with the constraints on its utilization.

Cooperation

Research Cooperation:
- University of Stuttgart, Institute of Computer Science, Stuttgart (D);
- Instituto per la Ricerca Scientifica e Technologica IRST, Trento (I);
- Georgia Tech University, Atlanta, Georgia, (USA);
- VTT Electronics, Oulu (SF);

Industrial Cooperation:
- Tecmath GmbH, Kaiserslautern (D)

Jean-Francois Girard

Minna Mäkäräinen

Cleanroom Software Engineering Group (A): Systematic Development

One of the goals of Software Engineering is the development of high software quality within reasonable time. Cleanroom provides one possible path to reach this goal. It combines pragmatic process principles with a sound theoretical basis to yield success in practice.

Cleanroom provides a complete process for the development of large software systems in such a way that the quality and reliability of the software produced is under control and certifiable. The process itself is comprised of five core techniques that provide the engineering rigor and precision necessary for achieving quality results. These techniques are:

– formal specification and design
– incremental development
– stepwise refinements of specifications into code
– correctness verification of code
– statistical quality certification.

Goals

The main goals of this group are:

- Develop a deeper knowledge on how Cleanroom can be especially tailored to the needs of software organizations and what factors have to be considered when introducing Cleanroom.

- Develop models/procedures on how the essentials of Cleanroom (core techniques) can be integrated into existing development processes, perhaps one by one.

Both goals provide the enabling conditions for the successful introduction of Cleanroom into organizations.

Group Focus

In particular, this group focuses on:

- Integrating Cleanroom with the object-oriented developing paradigm. This includes ways for incremental development with objects, functional verification of object-oriented code, derivation of Cleanroom models from object-oriented analysis models, etc.

- Allowing for systematic support even for the very early phases of software projects. This entails the integration of requirements engineering and requirements elicitation techniques into Cleanroom.

- Evaluating tools which support the core techniques of Cleanroom in order to assess their cost/benefits and applicability.

- Running controlled experiments and case studies in order to understand the restrictions and limitations of Cleanroom.

Scientific Issues

- How to generate usage models directly from the specification?

- How to adapt Cleanroom certification to platforms, operational uses and testing?

- How to incorporate domain-specific information into the development approach?

- How to make use of Cleanroom for reengineering tasks? This includes the mapping of Cleanroom concepts onto reengineering concepts and process definitions by using Cleanroom elements.

Practical Use

Cleanroom is a full-fledged systematic software development process. Any software organization integrating the major Cleanroom concepts or adopting the complete process can develop their software with guaranteed reliability and hence is one capable of keeping their projects under intellectual control. The latter entails that development control and product correctness are concurrently maintained.

Cooperation

Research Cooperation:

– Dycon Systems, Bethseda, Maryland (USA)
– Software Engineering Technology Inc. (SET), Knoxville, Tennessee (USA);
– University of Kaiserslautern, Kaiserslautern (D);
– University of Tennessee, Knoxville, Tennessee (USA);

Industrial Cooperation:

– Q-Labs Software Engineering GmbH, Kaiserslautern (D)

Christian Bunse

Stefan Jungmayr

Cleanroom Software Engineering Group (B): Software Inspection

Despite important changes made to software development practice since the term software engineering was coined thirty years ago, software quality deficiencies and cost overruns continue to afflict the industry. One of the major problems is that each of the different steps found in today's development life cycles is prone to introduce new defects. This is true from the problem all the way to its coding addressed by the system.

Software inspections, developed at IBM by Fagan in the early 1970s, can be used to tackle this problem, though they are only being introduced very slowly. Inspections enable the detection and removal of defects after each process phase. In this way, up to 80 percent of all software defects can be identified and eliminated early during software development. Hence, inspections can have a resounding effect on reducing rework cost and delivery time, because defects do not slip through various development phases. When defects do slip, it gets increasingly expensive and time-consuming to remove.

Cost and schedule reductions for typical applications have been shown to be on the order of 30 percent when a good inspection process is used. In order to be most beneficial, an inspection should follow a well-defined process including three major steps: Defect detection, defect collection, and defect correction.

Throughout the defect detection and collection step, a team of technical personnel analyzes the software product with the help of a reading technique for defects detection. They then agree on a set of defects, thus eliminating false positives, and rank the faults with respect to their severity level. Defect correction sees the product's author suitably eliminate the defects.

Goals

The main goals of this group are to

- provide supporting techniques and tools for each single inspection step.

- introduce, perform, and analyze existing inspection techniques within organizations.

Both goals allow practitioners and researchers to identify the most suited and tailored inspection approach for particular organizations.

Group Focus

In particular, this group focuses on

- identifying and understanding the factors that have an influence on inspections,

- performing controlled experiments to find out how to tailor inspection techniques to a given context within an organization,

- technical and tool support for the defect detection steps in the form of well-defined, tailored reading techniques,

- decision guidelines about which of the existing inspection techniques appear to work best in practice, both from a scientific and a practical point of view.

Scientific Issues

- There is an ongoing discussion in the literature on whether or not defect detection is a group activity and therefore should be conducted in meetings. There is evidence to suggest that performing inspections with meetings are as beneficial as performing inspections without meetings. However, this needs further investigation and, more importantly, further validation.

- Although inspection is considered a human-based technology, some of the steps may be supported by tools. The question is, to what extent can a tool contribute to inspection success.

- Although inspections are clearly beneficial because they can detect defects early in the software development process, they also incur costs. First, they have a delaying effect within that process; second, some corrected defects may not have been defects proper; and third, new defects may be inserted. All three cost drivers need to be balanced against the benefits of inspection to clearly understand the return investment from the latter.

- Research in inspection is highly experimental in nature. Some of the questions regarding inspection must be answered by collecting and analyzing inspection data. Examples are the following:

  – How many defects are still in the document after it has been inspected?
  – How many participants are necessary to detect the optimal number of defects?
  – What is the return on investment for inspection (cost/benefits ratio)?

Practical Use

Inspections are widely used in software organizations where quality and productivity are both critical to a company's survival. Inspections are an industry-proven best practice for software quality improvement and cost reduction. Following an up-front investment, the introduction of inspections in a project typically pays off in an increase in efficiency. This can be translated into schedule reduction. Considering the positive impact inspections have on a project, it is not surprising that the interest in inspections and reports of successful results have been strongly increasing since its development in the 1970s.

Cooperation

Research Cooperation:
– University of Kaiserslautern, Kaiserslautern (D);
– University of Maryland (USA)

Industrial Cooperation:
– AEG Atlas GmbH, Frankfurt/Main (D);
– Allianz Lebensversicherungs-AG, Stuttgart (D);
– Daimler-Benz Aerospace DASA, Bremen (D);
– Daimler-Benz AG, Forschung und Technik, Ulm (D);
– Ericsson Eurolab Deutschland GmbH, Herzogenrath (D);
– Robert Bosch GmbH, Frankfurt/Main, Stuttgart (D);
– Siemens AG, ZFE, München (D);

Christian Bunse

Oliver Laitenberger

Requirements Engineering Group

Developing systems that fulfill the needs of the customer to the maximum possible extent is at the core of any software engineering project. Hence these needs first have to be clarified and well-documented, and this is the role of Requirements Engineering (RE). Otherwise, systems do not meet customer requirements, are difficult to maintain, and/or cost and schedule overruns are a likely result.

Since Requirements Engineering is the interface between customer and developer, it plays a key role in the overall success of the software development project. It has three major phases: elicitation, modeling and validation/verification.

Requirements elicitation addresses gathering requirements from stakeholders and is a multi-disciplinary process involving managers, end-users, software developers, and maintenance stake-holders. As a social process it is burdened with all the delicacies of human interaction.

Requirements modeling deals with the production of the requirements documents. As these documents form the basis for all further software development phases, any errors introduced at this stage have a critical impact on software development and will be costly to repair. They also influence project management, since the requirements form the basis for cost and schedule estimates.

Validation ensures that the various stakeholders reach a consensus about the elicited set of requirements. Verification deals with the proof of essential properties like consistency, completeness, and safety-relevant invariants.

Goals

The goal of this group is to define requirements engineering processes and product models which

- support communication between end-users and software developers
- define precisely the required semantics and properties of the system that is to be developed
- lead to early detection of errors in the requirements, while trying to prevent them in the first place
- give a basis for future maintenance and evolution of the resulting system.

Processes and products will usually depend on the characteristics of the development organization and the type of application to be constructed.

Group Focus

The RE group concentrates on the following issues:

- In the case of real-time, embedded systems, how can one construct requirements documents that satisfy the desired qualities? Especially, how can one formalize requirements to circumvent ambiguities and how can these formalized requirements be communicated to stakeholders?

- Evaluating a number of formal description techniques, in particular the SCR (Software Cost Reduction) requirements technique.

- Due to the large number of persons involved in requirements engineering and the diversity of their goals, the negotiation of requirements is very important in order to achieve general acceptance of the final system. How can this process be adequately structured and enacted in order to achieve this aim?

- Requirements engineering for a family of systems, that is of systems that deal with the same application domain, is usually studied independently of the analysis of single application systems. How are these activities different and what can both activities learn from each other. How does domain analysis impact the elicitation and modeling processes?

- Requirements engineering is influenced by the software engineering process, customer relations, type and size of the application. Consequently, the characteristics required of the requirements engineering process and products will vary. Can this dependency be adequately described and if so, can it be used to derive prescriptive rules for tailoring process and products?

Scientific Issues

- The various stakeholders will usually have different views on the characteristics required from a system that will adequately address their needs. These views need to be elicited and reconciled into a common view. This process needs to be supported by effective means, such as tools or adequately-structured social processes.

- The requirements documents form the basis for product development and maintenance. Thus their quality has a major impact on these activities. Consequently, maintaining the quality of these documents has to be an ongoing concern during their development and needs to be finally ascertained by means such as inspection or static analysis.

- The reuse of artifacts during software development is a key in producing high quality products with little costs. The higher the abstraction level of the artifacts, the larger the potential gains. Thus it is an important issue to find ways for supporting the systematic reuse of requirements during the application development process. This is especially true when developing a family of systems.

Practical Use

Requirements engineering is a critical step that must precede any software development effort. It is key to commercial success of a project and to customer satisfaction as it is necessary to ascertain that the system to be built will meet customer needs and that the customer understands the implications of fielding the system in its overall business environment.

To illustrate the importance of having a good requirements engineering process, it has been demonstrated that it costs 5 to 20 times more on average to remove a defect during implementation than during the requirements engineering phase.

Cooperation

Research Cooperation:

- University of Kaiserslautern, Kaiserslautern (D);
- McMaster University, Hamilton, Ontario (CAN);
- Naval Research Lab, Washington, D.C. (USA)

Industrial Cooperation:

- Bosch Telecom GmbH, Frankfurt/Main (D)

Erik Kamsties

Klaus Schmid

Industrially-funded Projects

Industrially-funded projects are the core business of the IESE: Projects are designed to take care of the special needs of the customers. They depend upon the size of the customer and the type of department - r&d, software development, quality assurance - with which we collaborate. Industrial projects thus vary along different dimensions:

– They vary between pure technology transfer and pure r&d.

– They can be short-term or long-term.

– They can focus on directly increasing developers' know-how or on creating leveraging competence that enables the customer to self-improve.

Especially when canvassing long-term collaborations, we design a series of projects rather than one monolithic project. We start with an expert-study, a workshop and customer-specific trainings and then move on to more long-term improvement programs.

Dr. Frank Bomarius, Department Head

Collaborations within industrially-funded projects

Partner	Cooperation Style
Allianz Lebensversicherungs-AG, Reinsburgstraße 19, 70178 Stuttgart	Technology Transfer
Bosch Telecom GmbH, Kleyerstraße 94, 60326 Frankfurt	Technology Transfer
Daimler Benz Aerospace, Produktbereich Raumfahrt-Infrastruktur, Postfach 105909, 28059 Bremen	Technology Transfer
Daimler-Benz AG, Forschung und Technik, Wilhelm-Runge-Straße 11, 89081 Ulm	Research and Development
Ericsson Eurolab Deutschland GmbH, Ericsson Allee 1, 52134 Herzogenrath	Research and Development
Kommunikations- und Datentechnik GmbH, Rottendorfer Straße 7a 97072 Würzburg	Technology Transfer
Markant Südwest GmbH, Winzler Straße 152-160, 66955 Pirmasens	Technology Transfer
Q-Labs Software Engineering GmbH, Sauerwiesen 2 67661 Kaiserslautern	Technology Transfer
Robert Bosch GmbH, Kleyerstraße 94, 60326 Frankfurt	Technology Transfer
Societa Interbancaria per l´Automazione, Viale Certosa, 218	Research and Development

AMOS - Assessment Model for Object-Oriented Software Architectures

Objective

The objective of this project is to work out a model for the assessment of the quality (reliability) of the object-oriented software architecture developed within a pilot project.

Approach

In the project, a measurement program is set up to collect data which helps to build the intended models.

First, change report data about every change that is made to the system, whether it be as a result of changing requirements or as a result of an error, is recorded.

Second, the software architecture is assessed by means of the most appropriate object-oriented coupling, cohesion and inheritance measures available in the literature.

Each measure is empirically validated - that is, the relationship between each measure and the quality (i.e., reliability as measured by the change report data) of the software classes in the architecture is evaluated. Measures which are found to be good indicators of software quality are then used to build a predictive model of software quality for this architecture.

Results

A quality model is built from those measures found to be the most significant predictors of the reliability of the software classes with respect to defect distribution. This model can then be used by developers to plan future resource allocation for inspections and testing predicted defect types.

In addition, the developer effort distribution found in this project will enable identification of areas where not enough effort has been spent on the development life-cycle which could have influenced the resulting software architecture, e.g., too much effort spent proportionately on implementation and testing as compared to design and inspections.

Partner:
Daimler-Benz AG,
Forschung und Technik
PO-Box 2360
D-89013 Ulm

Contact
Dr. Lionel Briand
email: briand@iese.fhg.de
Tel: +49 631 707 251
Fax: +49 631 707 202

Cost estimation model for sd&m

This study was performed by the Fraunhofer IESE between September 1996 and March 1997 for software design & management GmbH und Co. KG (sd&m), a software development company in Munich.

Objective

The objective was to analyze the cost estimation procedures in effect at sd&m and to build an explicit sd&m-specific model for reliably estimating the cost of future projects. This model enables less experienced project managers to make reliable, repeatable, and justifiable estimations and relieve experienced managers from the burden of being involved too frequently in cost estimations. It also allows managers to perform a cost-related risk assessment at the start of a project and plan for appropriate steps to reduce risk.

Approach

All analysis performed during the study was based on information gathered in a series of interviews from some of the most experienced sd&m project managers. Each step of analysis as well as synthesis of the model was reviewed and verified by sd&m managers. The cost estimation model proper was validated using project data from successful sd&m projects.

The cost estimation model developed during the study is particularly tailored to the type of projects and the style of project management at sd&m. Strictly speaking, the model is applicable only in this particular context. The core of the model is based on a specific subset of the most important cost factors identified for sd&m during the study.

...predict and control cost of projects ...

The model was validated and adjusted using data from ten past sd&m projects. This data provided the benchmark for cost estimation in future projects.

From the point of view of the user, i.e., the project manager, the model itself and the benchmark can be considered a black box. To use the model, the user deals with input data to the model, that is, a set of project data (estimates) collected by means of a project data questionnaire, and the output of the model, which is the cost estimate.

Each of the cost-driving factors considered by the model contributes to this estimate. Depending on the project under consideration, i.e., depending on the actual input from the project data questionnaire, some factors contribute more than others to the estimate. In other words, for the given project some factors have a higher association with cost than others.

To assess the level of risk and consequently reduce the risk of cost, those factors that contribute most to the estimate must be identified. Therefore, besides the cost estimate itself, the model also provides a ranking of the factors with respect to their relative association with cost.

Results

The study produced a validated cost estimation model. The output of the model can be used to determine the risk level of the project and, depending on the risk level and the highest ranked cost-driving factors, the project manager can now select a set of risk reduction procedures that are suitable to lower the risks.

Partner:
sd&m software design & management GmbH und Co. KG
Thomas-Dehler-Straße 27
D-81737 München

Contact
Dr. Frank Bomarius
email: bomarius@iese.fhg.de
Tel: +49 631 707 121
Fax: +49 631 707 203

Efficiency Improvement of Software Development at Space Infrastructure Division of Daimler-Benz Aerospace AG

The continuously growing software portion in products and services in the aeronautics and space industry is of paramount importance for the systems built. Demands for this type of software, such as mission criticality, complexity, or sheer size, are increasing.

At the same time, Daimler-Benz Aerospace (DASA) is exposed to substantial budget cuts in European space programs. Consequently, only fixed-price contracts are concluded by the European Space Agency (ESA) and competition among providers increases. To remain competitive and satisfy customers, DASA launched improvement programs that are targeted at keeping tight schedules and at improving productivity and quality.

At the Department Informatics and Operations of the Space Infrastructure Division at DASA, improvement goals have been identified in the course of an analysis performed in early 1996 by the Fraunhofer IESE and other external consultants. The analysis aimed at restructuring the internal business processes.

Objective

The cooperation with Fraunhofer IESE has been launched to tackle the improvement goals related to software development, namely to:

(1) Improve the transparency of the software development process.

(2) Detect defects earlier in the software development process, reduce cost, increase reliability and predictability.

(3) Capture and prepare experience from software projects for reuse in future projects and across application domains.

Approach

The general approach pursued in this cooperation is to improve the software process by means of measurement-controlled process improvements. The measurement program was set up and performed according to the Goal/Question/Metric-Approach (GQM).

During an initial training phase, the development process of the software component CSS (Core Simulator Software) was selected as pilot project. The CSS simulator supports the software development for Columbus, the European contribution to the space station Alpha. Within this project a GQM measurement program was set up to analyze the project in retrospect.

Pertaining to goal (1),
the measurement program has been expanded to new projects in order to gain experience in setting up measurement programs and to increase transparency.

Driven by goal (2),
new software techniques, namely inspection techniques, have been selected, customized to the particular needs of DASA, and introduced to the software development process. Inspections are applied to software development artifacts of early development stages in order to decrease defect rate, increase reliability and predictability, and cut costs for rework. Their performance is monitored by means of a tailored measurement program.

Results

A support team has been established at DASA that is now taking on the task of defining, setting up, and performing controlled improvement steps. In particular, this team is now ready to set up and perform measurement programs and to perform inspections.

The measurement results captured so far are viewed as a first step in the long-term goal of establishing an Experience Factory at DASA. An Experience Factory tool is currently being set up to capture experience gained so far.

Partner

Daimler-Benz Aerospace AG
Space Infrastructure Division
PO-Box 105909
D-28059 Bremen

Contact

Dr. Frank Bomarius
email: bomarius@iese.fhg.de
Tel: +49 631/707-121
Fax: +49 631/707-203

MS WWS 2000 - Markant Südwest Merchandising System

Project MS WWS 2000 is a software development effort of Markant Südwest AG, a trading company. Markant used a commercial merchandising system in the past and has decided in 1995 to implement a new system on its own which fits its needs better than currently available commercial products and does so at a lower cost.

Objective

The short-term goal is to deliver the merchandising system needed by Markant without compromising its scalability and variability. On a longer-term basis, a generic, domain-specific architecture and its associated reusable component will be developed so as to enable this merchandising system to adapt to different lines of merchandising activities.

The principal role of the IESE within this project is the establishment of an efficient and highly-productive software development team at Markant. This means providing software engineering techniques and methods that meet the particular needs of Markant. Furthermore, the IESE supports Markant in domain analysis, requirements engineering, the selection of techniques, methods and tools, overall system design, and in the management and optimization of the software development process by means of coaching and training.
This year, Markant Südwest Software- und Dienstleistungs GmbH decided to locate its office within walking distance of the IESE to benefit from a strong, daily interaction with us.

Approach

The project started with a domain analysis and initial optimizations of organizational structures. Requirements analysis and specification were conducted using an adaptation of Booch's methodology using Rational Rose (TM).

For the system modeling phase the Booch method was also employed. As the implementation phase proceeds a description of current practices of software development is produced. This description is the prerequisite for goal-oriented measurement programs and improvements which are in the process of being introduced.

Continuous process optimization according to the improvement goals and preparation of ISO 9000 certification of Markant's matured software development process will be one of the IESE's future roles within this project.

Results

The software development team has been assembled and trained in the relevant methods and techniques. The architecture of the WWS2000 system has been determined and a detailed design has been worked out. The new merchandising system will use client-server technology. The client side is object-oriented, while the server side is a conventional high-performance relational database system.

For the implementation of the client-part which comprises the GUI, the application logics, and the object-relational mapping, VisualAge (IBM) is used. The server-part is on an IBM AS/400 accessed through a network via SQL.

Work has now started to transition the WWS2000 architecture to the generic merchandising system architecture. The result of this work is expected to begin system instances generation in 1998.

The system is to be operational in early 1998.

Partners

Markant Südwest Handels AG
Winzler Straße 152-160
D-66955 Pirmasens

Markant Südwest Software- und Dienstleistungs GmbH
Sauerwiesen 6
D-67661 Kaiserslautern

Contact

Dr. Jean-Marc DeBaud
email: debaud@iese.fhg.de
Tel: +49 631 707 251
Fax: +49 631 707 202

PRIME - Predicting Inspection Metrics

Today, the market for telecommunications products is rapidly growing while at the same time, competition is increasing. For Ericsson, as a major global player in this market, it is important to reduce time-to-market, to lower development costs and improve quality aspects such as reliability.

Since the proportion of software to the value of telecommunication products is up to 80%, the software development process and the techniques and methods employed are crucial for Ericsson's success.

Project PRIME focuses on software inspections at Ericsson. Inspections are an important means of reducing costs and increasing reliability of software products, if they are used in early phases of the software development process.

Objective

The goal of project PRIME is to show that the use of quantitatively-controlled inspection methods helps to make significant improvements within the software development process at Ericsson. It is expected that this will enable Ericsson to improve control over the inspection process and to assess the quality of the inspected documents.

Approach

During a pre-study, the existing inspection process at Ericsson was analyzed, inspections were attended, and inspectors were interviewed in order to describe the current inspection process.

Then the description of the inspection process was used to develop capture-recapture models. Such models allow the prediction of the number of defects remaining in a document after it has been inspected.

Both the model of the inspection process and the prediction model are then validated with historical data from Ericsson's projects. Based on the results, formal inspection methods are evaluated and introduced experimentally in a project. This helps improve the effectiveness and efficiency of the inspection process and assess the prediction model in practice.

Results

There are three major results of this project:

- A description of a well-defined inspection process as it is used in reality within Ericsson.

- A well-defined procedure for collecting inspection data. This includes a procedure for inspection data analysis.

- A prediction model to assess the quality of the inspected document.

Proper application of the inspection process together with a validated model for inspection data analysis and prediction of inspection effectiveness provide Ericsson with a powerful tool to control the quality of the software product and to lower costs.

Partner

Ericsson Eurolab Deutschland GmbH
Research Department
Ericsson Allee 1
D-52134 Herzogenrath

Contact

Dr. Frank Bomarius
email: bomarius@iese.fhg.de
Tel: +49 631 707 121
Fax: +49 631 707 203

PROMO - Practical Guidelines for Process Modeling

The purpose of the PROMO project is to refine applicable guidelines for process modeling in the context of systematic and continuous process improvement at Daimler-Benz AG.

The first version of the process modeling guidelines has been piloted in two projects at Daimler-Benz AG, the one at AEG Atlas GmbH (software systems for energy distribution) and the other at Mercedes-Benz AG (software systems for automotive electronics).

Objective

The goal of the PROMO project is to collect the experiences the pilot projects had in using the process modeling guidelines, and to revise the guidelines accordingly. The revised process modeling guidelines will be employed in projects at Daimler-Benz AG.

Approach

The project starts by collecting the experiences of the pilot projects in using the process modeling guidelines. Two methods are used:

First the persons involved in the pilot projects write down their experiences about using the guidelines, and then the experiences are discussed in a meeting with the persons involved in pilot projects and the PROMO personnel. From the experiences, practical proposals for improving the process modeling guidelines are derived, and the guidelines are revised accordingly.

The process modeling guideline follows the NASA/SEL Quality Improvement Paradigm (QIP). The process modeling formalism used in the pilot projects was slightly adapted from IDEF0 formalism.

Results

As a result of the project, practical experience on the requirements for process modeling guidelines is gained, in particular, applying process modeling methods in real-life projects, and the kinds of benefits that can be achieved when applying process modeling.

As an intermediate result of the project, a summary of the experiences of the pilot projects and improvement proposals was derived from the experiences. The final result of the project is the revised version of the process modeling guidelines, which can be used in future projects at Daimler-Benz AG.

Partner

Daimler-Benz AG, Forschung und Technik
PO-Box 2360
D-89013 Ulm

Contact

Dr. Lionel Briand
email: briand@iese.fhg.de
Tel: +49 631 707 251
Fax: +49 631 707 202

Systematic Improvement of Software Quality by Goal-Oriented Measurement at Allianz Lebensversicherungs-AG

With the advent of an open market in the European Community the quality of software development processes and their resulting products has gained high impact on the competitiveness of insurance companies all over Europe. This development has been recognized early by Allianz. As a preventive action a program for the systematic improvement of software quality was established at the Allianz Lebensversicherungs-AG Stuttgart. The program was started in 1993 and was supported by STTI-KL until the end of 1995. Since 1996 the Fraunhofer IESE continues to conduct the program.

Objective

The objective of the program is to increase the competitiveness of Allianz in the European insurance market by means of improving quality and productivity and time-to-market of software development, thus, allowing timely reactions as new marketing opportunities arise.

Approach

The cornerstone of this project is the Quality Improvement Paradigm (QIP), a modification of Total Quality Management (TQM) with special emphasis on software development.

The QIP as conducted by the Fraunhofer IESE is closely related to goal-oriented software measurement. The implementation of this paradigm provides an organizational infrastructure called Experience Factory (EF) to capitalize and reuse life-cycle experience and products.

The first step in this project was the definition and implementation of a goal-oriented measurement program for a set of pilot projects. This step provided a baseline of measured data reflecting the state of the software development processes and products at the time the project was started.

Improvement from subsequent steps can be measured quantitatively referring to these baseline figures.

Software quality is continuously improved by systematic use of efficient review and inspection techniques. Ongoing measurement is used to monitor the achievements.

The development of a prototype experience base allows the reuse of software measurement plans. Furthermore it enables corporate learning in the long run. It makes experiences explicit and turns them into corporate legacy.

Results

The level of understanding regarding the software development process has increased in many respects, such as effort distribution, amount of rework, reasons for rework, relevant influence factors on the stability of requirements, amount and scope of changes.

Hypotheses that existed or had been stated at the beginning of the program have been confirmed or have been refuted by means of quantitative methods.

The project has clearly identified problem areas, such as overvaluing testing as a means of creating quality.

Areas of potential improvement have been discovered. For instance, it was learned that the level of experience has much more influence on the effort spent than the use of CASE tools.

Software engineering know-how has been improved in many ways. Most importantly, goal-oriented measurement applied to software processes and products has enabled the software developers to continuously self-improve in a well-controlled way.

Partner
Allianz Lebensversicherungs-AG
IS/Verfahrenstechnik
Reinsburgstraße 19
D-70178 Stuttgart

Contact
Dr. Günther Ruhe
email: ruhe@iese.fhg.de
Tel: +49 6301 707 121
Fax: +49 6301 707 203

Publicly-funded Projects

Collaborations exist with many publicly-funded consortia aimed at either software engineering technology advancement or dissemination of best practices. Publicly-funded projects can be devoted to both research and development and technology transfer. Additional bilateral industrial collaborations often result from performing these projects. In 1996, the projects shown in the list below were performed:

Project	Partner	Funded by	Title	Start
Softquali	Daimler-Benz (F&T) (D) AEG Atlas GmbH (D) Allianz Lebensversicherungs-AG (D) Simens AG (D)	German Ministry for Education and Research (BMBF)	Systematic Improvement of Software Quality by Goal-Oriented Measurement and Explicit Reuse of Software Know-How	5/1/1995
Muvie	GfAI (Gesellschaft zur Förderung angewandter Informatik e.V.), (D) AIF (Arbeitsgemeinschaft industrieller Forschungsvereinigungen „Otto von Guericke“ e.V.) (D)	German Ministry of Science (BMWi)	Graphic Support of Complex Business Processes Using Multiview Editors for Workflows	1/10/1995
Midas	Societa Interbancaria per l´Automatione (I) CEFRIEL (I)	CEC ESPRIT/ESSI Project No. 21244	Measurable Improvement of Development, Deployment and Operation of Interbank Automation Software	1/1/1996
CEMP	Robert Bosch GmbH (D) Schlumberger RPS (NL) Digital Equipment SPA (I) CEFRIEL (I)	CEC ESPRIT/ESSI Project No. 10358	Customized Establishment of Measurement Programms	1/1/1994
PERFECT	Daimler Benz AG (D) Robert Bosch GmbH (D) Siemens AG (D) Q-Labs Software Engineering GmbH (S) Cap Gemini Innovation (I) Laboratoire LSR (F)	CEC ESPRIT/ESSI Project No. 9090	Process Enhancement for Reduction of Software Defects	1/10/1993
PROFES	Dräger Medical Electronics (NL) Ericsson (Fi) Etnoteam (I) Schlumberger (F) University of Oulu (Fi) VTT Electronics (FI)	CEC ESPRIT/ESSI Proposal No. 23239	Product-Focused Improvement of Embedded Software Processes	1/1/1997
STTI Kaiserslautern	Q-Labs Software Engineering GmbH (D) Markant Software- und Dienstleistungs GmbH (D) Tecmath GmbH (D) ICON Intelligent Control Gebäudetechnik GmbH (D) Schönfisch & Faust Computer Integration (D)	Ministry of Economic Affairs, Transportation, Agriculture and Viniculture of the State of Rhineland-Palatinate, Germany	Support of Small and Mediumsize Enterprises	1/1/1996

Publicly-funded projects play an essential role in the research and technology transfer strategy of the Fraunhofer IESE. They are used to

- collaborate with leading research institutes in projects of strategic relevance,
- cooperate with industrial partners to develop innovative solutions for their problems,
- transfer technologies into industry and establish related know-how.

There are different forms of publicly-funded projects corresponding to the funding organization and the scope of the funding program. Currently, five avenues of public research projects are maintained at the Institute:

- European ESPRIT projects
- European ESSI projects
- German BMBF projects
- German AiF projects
- Projects funded by the State of Rhineland-Palatinate

ESPRIT (European Strategic Programme for Research and Development in Information Technologies) is a European program designed to ensure that Europe's industries gain competitive advantage from efficient use of communication and information systems. Within the fourth framework of the ESPRIT program the Fraunhofer IESE was involved in two international R&D projects in the area of 'Software Intensive Systems Engineering': PERFECT and PROFES. Both projects were used to collaborate with leading research and industrial organizations. The general objective was to develop competitive know-how for process and product improvement which can be used in subsequent industrial improvement programs.

The goal of the European Systems and Software Initiative (ESSI) is to promote improvements in the software development process in industry, by taking up well-founded and established - but insufficiently developed - methods and technologies, so as to achieve greater efficiency, higher quality, and greater economy. Fraunhofer IESE was involved in a Process Improvement Experiment aimed at demonstrating software process improvement in the configuration management domain. The MIDAS project will be described subsequently.

In 1994, the German Federal Ministry for Research and Technology (BMBF) initiated a special program to support software technology. The objective of the program was to increase the competence of German industry in software development. Among the few projects that were selected for funding was the SoftQuali project which is described in more detail in this chapter.

The Ministry of Economic Affairs, Transportation, Agriculture and Viniculture of the State of Rhineland-Palatinate supported the dissemination of process and software engineering technologies to small and medium size companies in Rhineland-Palatinate.

The 'Arbeitsgemeinschaft industrieller Forschungseinrichtungen e.V.' (AiF) is a German organization for industrial collaboration. Special emphasis of this program is in small and medium-size enterprises. Together with the 'Gesellschaft zur Förderung angewandter Informatik' (GFaI), Fraunhofer IESE participated in a project called MUVIE.

Dr. Günther Ruhe,
Deputy Director, Department Head

Measurable Improvement of Development, Deployment and Operation of Interbank Automation Software (MIDAS)

The Societa Interbancaria per l´Automazione (SIA) is in charge of running, developing, and maintaining the national interbank network of Italy. Reliability and availability of interbank services offered by SIA is of essential importance to all the financial transactions performed within this network. Effectiveness of configuration management is, in general, expected to be of essential importance to the quality of the corresponding software development. However, no detailed qualitative and quantitative information was available about the main factors that influence successful performance of configuration management, and about ways to exploit this information for optimal project performance.

Approach

Improvement was achieved by baselining the SIA software process and establishing an effective configuration management process. A suitable measurement program was defined and conducted in order to objectively assess the effectiveness of the new configuration management practice. Configuration Management Definition of the configuration management process covers description of included policies, roles, and tools. The process modeling activity started from the problem reporting, tracking, and solving activities, and then was extended to the whole software life-cycle. Implementation of the process involved modification and optimization of the initial model and determination of critical success factors.

The effectiveness of the configuration management policies and tools was assessed by means of a measurement program which is based on the Goal/Questions/Metrics (GQM) paradigm. It contains the definition of GQM and measurement plans. The subsequent comparative analysis of measurement data reflected the situation before and after the introduction of configuration management in terms of cost and benefit.

Results

- Definition of the configuration management process covering the description of included policies, roles, and tools. The process modeling activity started from the problem reporting, tracking, and solving activities, and was extended to the whole software life-cycle.

- Introduction of goal-oriented measurement within SIA.

- Stepwise definition of measurement goals by preference modeling.

- Definition of the GQM plans to evaluate benefits of configuration management.

- Modeling of configuration management processes.

- Execution of the measurement plan to assess the original situation (with respect to the baseline project). This involves the collection and analysis of data like the number of detected problems, the service availability time, the effort employed to correct errors, etc.

- Performance of common feedback sessions for analysis and interpretation of results.

- Experience packages on the above topics.

Partner

SIA - Societa Interbancaria per l´Automazione
Viale Certosa, 218
20156 Milano, Italy

Contact

Dr. Günther Ruhe
email: ruhe@iese.fhg.de
Tel: +49 6301 707 121
Fax: +49 6301 707 203

MUVIE
Graphical Modeling of Business Workflows with Multi-View Editors

Graphical modeling of flows and structures helps managers and engineers to understand and communicate systems in many disciplines. In practice, the applicability of visual representations is limited because the design of large systems results in huge and complicated graphics.

Approach

The objective of the MUVIE project is to research on Multi-View Design Environments, dealing with the complexity of graphical representations maintaining user defined views on the graphics.

Previous projects have shown, that simple approaches like the Multi-View-Controller approach are not sufficient in this case.

In MUVIE, each view defines a focus on the underlying graph structure, visualizing only those parts of the system that pertain to the view. Incremental changes are managed by a formal approach called Graph Model. In this approach, all changes to the central graph structure (Central Abstract Graph) are mapped to graph replacements.

Because all views are mapped to subgraphs of the central graph structure, graph replacements can be used to update the views. Sophisticated user interaction techniques, including direct manipulation and hypertext navigation support helps to manage relations among different views. Adaptable layout algorithms are integrated and relieve the user from complex manual placement tasks.

Architecture

The MUVIE architecture implies an independent module for management of all interrelationships among views and artifacts which is called Multi-View-Engine. The Multi-View-Engine is realized as a separate process and works completely independent of the visual representation of all artifacts.

Changes are processed incrementally based on a formal Graph Model, which ensures scalability and openness.

Results

The major results of the MUVIE-Project are:

- A set of documented techniques for building integrated multi-view environments based on graph structures. The techniques include interactive direct-manipulation techniques which help the user to deal with multiple views and a conceptual framework for incremental change and update handling at the technical layer. The documents are collected in a final report and will be released as an internal IESE report.

- A prototypic implementation of a framework. The prototypic implementation encompasses a client/server approach which can be reused for building visual language environments, such as for example CASE- and hypertext tools. Components of the framework encompass support for interaction, visualization, automatic layout and management of views. The framework has been implemented in C++ both on Sun/Solaris and PV/WinNT platforms.

- A sample visual editor for FunSoft-Nets. As an evaluation of the prototypic framework, a concrete graphic editor for FunSoft-Nets has been implemented. This editor allows to define FunSoft-Nets from many different perspectives, to navigate between different representations.

Partner

GFaI – Gesellschaft zur Förderung angewandter Informatik e.V.;
IIEF - Institut für Informatik in Entwurf und Fertigung GmbH;
Ingenieurbüro Drews, Berlin;
Fraunhofer ISST, Berlin/Dortmund;
Elpro Leit- und Energietechnik GmbH;
REVIG - Rückstands und Emissionsvermeidungs Ingenieur-Gesellschaft mbH, Berlin

Contact

Dr. Peter Rösch
email: roesch@iese.fhg.de
Tel: +49 6301 707 121
Fax: +49 6301 707 203

PERFECT – Process Enhancement for Reduction of Software Defects

Systematic improvement in the software domain is based on basic concepts such as QIP (fundamental improvement paradigm for software development), experience factory (for organizational learning and reuse), and the GQM paradigm (for goal-oriented measurement). While these fundamental concepts are very convincing, their industrial application needs a more precise description that is supported by appropriate tools and augmented with industry-style introduction material.

Objective

The overall objective of PERFECT has been to assist European industry in the measurement-based improvement of software processes. A set of techniques, methods, and tools supporting the improvement activities has been developed.

Approach

The PERFECT Improvement Approach (PIA) guides the introduction and operation of company-specific process improvement programs. The PIA is defined through the following parts:

- Principles of systematic improvement.
- Generic models that make the improvement principles operational.
- A collection of refinements and instantiations of the generic models that provide operational support for systematic improvement.

The generic models are structured into three perspectives: a methodological perspective, an organizational perspective, and a functional perspective:

- The methodological perspective addresses the activities involved in systematic improvement.
- The organizational perspective addresses the roles and organizational entities involved in systematic improvement.
- The functional perspective addresses the organizational and human capabilities, as well as the tool support required for systematic improvement.

Results

The major project results are the PERFECT Improvement Approach, a handbook and tutorials about it, as well as software tools and environments. PERFECT has been structured into three work packages: methodology, platform, and applications.

The methodological result is the PERFECT Improvement Approach (PIA). It guides the introduction and operation of company-specific improvement programs based on the Quality Improvement Paradigm (QIP), the Goal/Question/Metric approach (GQM), the Experience Factory concept (EF), and process modeling. PERFECT has made contributions and provided support tools for each of these areas.

The platform tasks have developed tools and environments for supporting improvement programs, such as APEL and GQMaspect. APEL is a software engineering environment, integrating product management, process modeling and enactment, and measurement. GQMaspect is a GQM editor supporting the planning of measurement programs.

The applications have focused on and evaluated the methodology and platform developments in projects from the embedded systems and telecommunication domains. They have evaluated the PIA as very helpful and are now disseminating and spreading it to other projects and departments within their organizations.

ESPRIT Project No 9090

Partners

The PERFECT project was carried out by a European consortium within the ESPRIT program by:
Cap Gemini Innovation (F);
Daimler-Benz AG (D);
LSR: Logiciels, Systèmes, Réseaux Grenoble (F);
Q-Labs (S);
Robert Bosch GmbH (D);
Siemens AG (N);
Sintef (N);
University of Kaiserslautern/Fraunhofer IESE (D)

Contact

Dr. Günther Ruhe
email: ruhe@iese.fhg.de
Tel: +49 6301 707 121
Fax: +49 6301 707 203

PROFES – PROduct Focused improvement of Embedded Software processes

The increasing amount and complexity of software in embedded systems (like, e.g., telecommunications systems, medical instruments, retailing systems, or avionics) sets new requirements for the quality of the products as well as for the management of the development process. The amount of software-related work is often more than 70% of the development effort for the whole system, and the software has to be developed in very short cycles, taking into account its close relationship with hardware and other product technologies.

For competitive companies, the customer-perceived product quality is a driving force for the improvement of embedded software development. Existing improvement approaches, however, are neither tailored to the specific needs of embedded software development nor focused on product quality requirements. Often, improvement goals are mainly based on software development process maturity profiles resulting from software process assessments. Software process assessments, however, do not establish detailed links from domain-specific product quality characteristics to individual development process aspects.

Objective

The objective of the PROFES project is to support the embedded systems industry with a tailored improvement methodology that:

- focuses improvement actions on those elements of the software development process that contribute most to the critical product quality factors;
- combines and enhances the strengths of goal-oriented measurement, process assessment, product/process modeling and experience factory;
- is validated through case studies in three industrial organizations.

Approach

The PROFES product quality improvement methodology will be developed, validated and exploited in three parallel industrial case studies representing three different application domains for embedded systems. The industrial application partners Dräger, Ericsson, and Schlumberger have been selected based on a shared set of customer-driven product improvement goals. By integrating goal-oriented measurement, product/process modeling, reuse of experience, and an enhanced embedded systems process assessment approach, the PROFES methodology will link software-related product quality factors directly to software development process characteristics and enable continuous product-driven improvement.

Results

- PROFES methodology handbook, containing:
  - guidelines to the identification and usage of product/process relationships,
  - support of an integrated use of goal-oriented measurement, software process assessment (enhanced for embedded systems), and reuse of experience,
  - guidelines to business impact modeling;
- Tools to support the PROFES methodology;
- Presentation and training material;
- Packaged experience from the case studies, namely:
  - lessons learned from the application of the PROFES methodology,
  - cost/benefit models,
  - models describing specific relationships between software product quality factors and software development process characteristics.

ESPRIT Project No 23239

Partners

Dräger Medical Electronics (NL);
Ericsson Eurolab Deutschland GmbH (SF);
Etnoteam S.p.A. (I);
Fraunhofer IESE (D);
Schlumberger Retail Petroleum Systems (F);
University of Oulu (SF);
VTT Electronics (SF)

Contact

Dr. Günther Ruhe
email: ruhe@iese.fhg.de
Tel: +49 6301 707 121
Fax: +49 6301 707 203

SoftQuali

This project's main objective is to contribute to the theoretical and practical basis for systematic quality improvement in software industry. Quality improvement will be mainly based on

- goal-oriented measurement following the Goal/Question/Metrics paradigm,
- systematic review and inspection techniques,
- packaging and reuse of software best-practice know-how.

Objective

The focused-quality goals in the selected application domains are flexibility of software processes and reliability of software products. Beside concrete results for the involved companies, an essential goal is to discover commonalities and differences between the application domains. All investigations within SoftQuali are accompanied by cost/benefit analyzes. The whole project uses the experimental approach in conjunction with the Quality Improvement Paradigm as the underlying paradigm for systematic quality improvement.

Approach

Goal-oriented software measurement has proven to be as a crucial device in defining the current state and in deriving subsequent improvement actions in software development. The measurement process deals with the definition of goals, the derivation of metrics, the collection of data, their validation and analysis, and finally the interpretation of results in the context of the environment from which the measures were taken. Within the SoftQuali project, goal-oriented measurement is introduced for pilot projects at three sites (AEG Atlas GmbH, Allianz Lebensversicherungs-AG, Siemens AG).

Reviews and inspections belong to the most promising improvement techniques that can be applied to software development at all stages of the lifecycle for different artifacts such as requirements, design, or code documents. Applied from the very beginning, their application supports early identification of faults within the different phases of software development. Reviews and inspections will be introduced and investigated again in three parallel case studies.

Reuse of products, processes, and experience is a promising way of contributing to the development of high quality software. The concept of an Experience Factory institutionalizes the reuse of experience and supports:

- characterization and understanding (e.g., number of faults per component),
- evaluation and assessment (e.g., effectiveness of tool support),
- prediction and control (e.g., total project effort).

The project develops formalisms and methods to establish such experience packages.

Results

Technology transfer of goal-oriented measurement into pilot projects at the three experimental sites.

Performance of the measurement programs with investigation and comparative analysis of main factors influencing the quality aspects of reliability and flexibility.

Development and introduction of scenario-based reading techniques and validated results of their effectiveness and efficiency.

Guidelines and heuristics for the application of reviews and inspections in dependence of varying environments.

Evaluated prototypes for knowledge presentation, structuring, and reuse in the Experience Factory organization. Technology packages on goal-oriented measurement, reviews, and inspections, and related experience packages.

Partners

AEG Atlas GmbH;
Allianz Lebensversicherung AG;
Daimler-Benz AG;
Siemens AG;
Fraunhofer IESE

Contact

Dr. Günther Ruhe
e-mail: ruhe@iese.fhg.de
Tel: +49 6301 707 121
Fax: +49 6301 707 203

National Cooperations

University of Kaiserslautern, Kaiserslautern, Germany

Center for Learning Systems and Applications (LSA), University of Kaiserslautern, Germany

Institute of Computer Science, University of Stuttgart, Stuttgart, Germany

International Cooperations

Center for Advanced Empirical Software Research (CAESAR), University of New South Wales, Sydney, Australia

Centre de Recherche Informatique de Montreal (CRIM), Montreal, Canada

European Software Institute (ESI), Bilbao, Spain

Experimental Software Engineering Group of the University of Maryland (UMD/ESEG), College Park, USA

Federal University of Santa Catarina, Florianopolis, Brazil

Georgia Tech University, Atlanta, Georgia, USA

GrafP Technologies Inc., Montreal, Quebec, Canada

Instituto per la Ricerca Scientifica e Technologica (IRST), Trento, Italy;

Semantics Designs, Austin, Texas, USA

Software Engineering Technology Inc. (SET), Knoxville, Tennessee , USA

Software Engineering Institute (SEI), Carnegie Mellon University, Pittsburgh, Pennsylvania, USA

Software Engineering Laboratory (SEL), NASA/Goddard Space Flight Center, Greenbelt, Maryland, USA

Software Technology Transfer Finland, Espoo, Finland

University of Oulu, Oulu, Finland

University of Tennessee, Knoxville, Tennessee, USA

VTT Electronics, Oulu, Finland

International Software Engineering Research Network (ISERN); Coordinator of ISERN since 1996: Fraunhofer IESE

Members of ISERN:

CSIRO; Australia

Daimler-Benz Research Center; Germany

Fraunhofer Institute for Experimental Software Engineering; Germany

Lucent Technologies - Bell Laboratories; USA

Macquarie University; Australia

Nara Institute of Science and Technology; Japan

Norwegian University of Technology & Science; Norway

NTT Data Corp.; Japan

Quality Laboratories Sweden AB (Q-Labs); Sweden

University of Bari; Italy

University of Hawaii; USA

University of Kaiserslautern; Germany

University of Maryland at College Park; USA

University of New South Wales; Australia

University of Rome - Tor Vergata; Italy

University of Strathclyde; Scotland; U.K.

VTT Electronics; Finland

Visitors hosted

Prof. Dr. V. R. Basili, University of Maryland, USA, several visits

David M. Charrett, Macquarie University, Sidney, Australia, April 1-4, 1996

Valentina Plekhanova, Macquarie University, Sidney, Australia, April 28-30, 1996

Prof. Tereza G. Kirner, Federal University of Sao Carlos, Brazil, May 05 - June 02, 1996

Dr. Marc Kellner, SEI, Pittsburgh, PA, USA, June 13-14, 1996

Frank McGarry, Computer Sciences Corp., Lanham-Seabrook, MD, USA, June 17-28, 1996

Dr. David W. Hislop, Army Research Office, North Carolina, USA, June 21, 1996

Prof. Dr. David L. Parnas, McMaster University, Hamilton, Ontario, Canada, September 02, 1996

Prof. Dr. Thomas Seewaldt, Fachhochschule Mannheim, September 19, 1996

Prof. Dr. Jesse Poore, University of Tennessee, Knoxville, TN, USA, September 24, 1996

Robert Susmaga, Poznan University of Technology, Poznan, Poland, October 14-26, 1996

Prof. Dr. Conradi, University of Trondheim, Norway, October 15, 1996

Prof. Dr. Ross Jeffery, UNSW, Sydney, Australia, November 23 - December 03, 1996

Letters from Guest Scientists

Pierfrancesco Fusaro, Italy

Prof. Visaggio, head of the SER Lab (Software Engineering Research Laboratory), University of Bari, Italy, suggested that I should join the research group of Prof. D. Rombach at the University of Kaiserslautern. I was lucky. Prof. Rombach gave me the opportunity to start working in his newly-established institute in February 1996. I was one of the first foreigners at IESE and I cannot hide that I had some troubles adjusting to a completely new working and living environment. But thanks to the support of my colleagues, I was able to overcome the first difficulties and become acclimatized.

My main involvement has been in the MIDAS *) project. My research activities concern the fields of experimentation and the analysis of validity and reliability of SPICE-based assessments. SPICE is an international project that aims at delivering an ISO standard for software process assessment.

I think that the experience gained during this year will have a tremendous influence on my future career.

*) see page 52

Dr. Richard Webby, Australia

I was looking for a place to do a postdoctoral year abroad. My colleague Ross Jeffery, Professor and Head of the CAESAR research centre, recommended working in Kaiserslautern at Professor Rombach's new Fraunhofer Institute.

The working environment here at Fraunhofer IESE can be characterized as dynamic, young, and international. My main involvement has been in the SPEARMINT project, which is concerned with developing a practical approach and better tool support for software process modeling. I think I have been able to add my experience in user interfaces and object-oriented development to the team, and at the same time, learn much about the field of software process modeling. A spirit of open communication and cooperation prevails at IESE, which enables productive collaboration both within and between projects and groups.

CAESAR and IESE have already signed a joint agreement for formal collaboration. Upon my return to Australia in July 1997, my plan is to continue the SPEARMINT project as a joint initiative.

Minna Mäkäräinen, Finland

My group leader Markku Oivo at VTT in Finland recommended to me the newly-established Fraunhofer Institute for Experimental Software Engineering in Kaiserslautern, Germany. My year in Germany started in July of 1996 and is half-way over at the time I am writing this. Time surely flies!

Our international work environment is very inspiring. I have a French project manager, an American department head, German room-mate and my closest colleague is Canadian. So, the official language of the institute is, obviously, English.

I worked in the software maintenance and reengineering group in the department of Innovative Software Engineering. My research tasks included composing a large tutorial on software maintenance and building prediction models from maintenance data. I also worked on a customer project *), which was very interesting.

My time here has given me new, fresh viewpoints on my work. I am convinced I can use things I have learned here to make my research and practical work back in Finland richer and more versatile.

*) PROMO, see page 48

Professional Contributions

Lecturing Assignments at Universities

D. Rombach
Lecture:
Software Engineering I,
Department of Computer Science,
University of Kaiserslautern,
winter semester 1995/1996 and
winter semester 1996/1997

D. Rombach
Project Course:
Software Engineering I,
Department of Computer Science,
University of Kaiserslautern,
summer semester 1996

D. Rombach, Günther Ruhe
Lecture:
Software Engineering II,
University of Kaiserslautern,
Department of Computer Science,
summer semester 1996

D. Rombach
Project Course:
Software Engineering II,
Department of Computer Science,
University of Kaiserslautern,
winter semester 1995/1996 and
winter semester 1996/1997

Günther Ruhe
Lecture:
Experimental Software Engineering,
Friedrich-Schiller-Universität, Jena

Committee Activities

L. Briand:
PC Member and publicity chair, ICSM'96, Monterey, CA, USA

L. Briand:
PC Member, ICSE'96, Berlin, Germany, March, 1996

L. Briand:
General chair, WESS'96 - IEEE International Workshop on Empirical Studies of Software Maintenance, Monterey, CA, USA

D. Rombach:
General Chair, ICSE-18, Berlin, Germany, March 25-29, 1996

D. Rombach:
Chairman, Steering Committee ICSE (International Conference on Software Engineering), from 1996 to 1998

D. Rombach:
Member, Technologiebeirat, Rheinland-Pfalz, since 1994

D. Rombach:
Member, Supervisory Board of the German National Research Center for Information Technology (GMD), since 1996

D. Rombach:
Member, Advisory Board of Q-Labs, since 1996

D. Rombach:
Senior Member, Institute of Electrical and Electronics Engineers (IEEE), since 1996

C. Tautz:
PC member, Technology Transfer Workshop, ICSE 18, Berlin, Germany, March 1996

Journal Editorships

L. Briand:
Empirical Software Engineering: An International Journal

K. El-Emam:
Software Process Newsletter

D. Rombach:
IEEE Software Magazine

D. Rombach:
The Journal of Systems and Software

D. Rombach:
Informatik: Forschung und Entwicklung

D. Rombach:
International Journal of Software Process: Improvement and Practice

D. Rombach:
International Journal of Empirical Software Engineering

Presentations

F. Bomarius:
Introduction to Continuous Quality Improvement, Bosch Telecom, Frankfurt/Main, July 30, 1996

C. Bunse, F. Bernauer:
Objektorientierte Software-Entwicklung nach Booch, February 1996 (The tutorial was conducted for the Markant Südwest AG, Pirmasens, Germany)

C. Bunse:
Systematische Verbesserung von Software-Entwicklungsprozessen durch zielorientiertes Messen und Bewerten, Innovationsmarkt SteP, Hannover Messe Industrie, April 27, 1996

C. Gresse, I. Wieczorek:
Goal Oriented Measurement, held at the IESE on the occasion of the visit of KoDa GmbH, Kaiserslautern, Germany, June 24-25, 1996

D. Rombach:
Software Experience Factory, Ericsson Eurolab, Aachen, Germany, March 13, 1996

D. Rombach
Software-Entwicklungs-Know-how: Ein entscheidender Faktor für die zukünftige Wettbewerbsfähigkeit, DASA, Bremen, Germany, April 1996

D. Rombach:
Software Engineering für sicherheitskritische Anwendungen, TÜV-Köln, Germany, May 23, 1996

D. Rombach:
Software Experience Factory, Ericsson, Kaiserslautern, May 13, 1996

D. Rombach:
Kontinuierliche messbasierte Verbesserung von Software-Entwicklungs-Know-How, Seminar, Bosch Telekom, Backnang, Germany, November 26, 1996.

G. Ruhe:
Rough Set Based Data Analysis in Goal-Oriented Software Measurement, Allianz Lebensversicherungs-AG, Stuttgart, Germany, July 24, 1996

C. Tautz:
Introduction to QIP and the Experience Factory, Project meeting with KoDa at Fraunhofer IESE, June 24, 1996

C. Tautz:
Durchführung eines Messprogramms, DASA-RI, June 26, 1996

C. Tautz:
Prozessverbesserungsschritte, Schwerpunkt Inspektionen, DASA-RI, September 18, 1996

I. Wieczorek:
Zielorientiertes Messen in Software-Projekten, held at KoDa GmbH, Würzburg, Germany, August 12, 1996

I. Wieczorek:
Vorgehensweise bei Planung und Durchführung von GQM-basierten Messprogrammen, held at Robert Bosch GmbH, Stuttgart, Germany, December 17, 1996

Articles in Journals and Books

V. Basili, L. Briand, W. Melo:
How Reuse Influences Productivity in OO Systems, Communications of the Association of Computing Machinery, October 1996, Vol. 39, No. 10, pp.104-116

V.R. Basili, S. Green, O. Laitenberger, F.Lanubile, F. Shull, S. Sorumgard, M.V. Zelkowitz:
The Empirical Investigation of Perspective-Based Reading, Empirical Software Engineering, vol. 1, No. 2., pp. 133-164, 1996.

V. Basili, L. Briand, W. Melo:
A Validation of Object-Oriented Design Metrics as Quality Indicators, IEEE Transactions on Software Engineering, October 1996, Vol. 22, No. 10, pp. 751-761

L. Briand, K. El Emam, S. Morasca:
On the Application of Measurement Theory in Software, Journal of Empirical Software Engineering,1996, vol. 1, no. 1, pp. 61-88

L. Briand, S. Morasca, V. Basili:
Property Based Software Engineering Measurement, IEEE Transactions on Software Engineering, Vol.22, No. 1, pp. 68-86, January 1996

C. Bunse, P. Giese, M. Verlage:
Drei Ansätze zur Formalisierung informeller Beschreibungen von Software-Entwicklungsprozessen, GI Softwaretechnik-Trends, 16/2, ISSN 0720-8928, pp 28-33, May 1996

J. Daly, A. Brooks, J. Miller, M. Roper, M. Wood:
Evaluating Inheritance Depth on the Maintainability of Object-Oriented Software, Journal of Empirical Software Engineering,1996, Vol. 1, No. 2, pp. 109-132

C. Differding, D. Rombach:
Kontinuierliche Software-Qualitätsverbesserung in der industriellen Praxis, Software-Metriken in der Praxis, Christof Ebert, Reiner Dumke (Eds.), 1996

K. El Emam, N. H. Madhavji:
An Instrument for Measuring the Success of the Requirements Engineering Process in Information Systems Development, Journal of Empirical Software Engineering, 1996, Vol. 1, No. 3, pp. 201-240

K. El Emam, D. R. Goldenson:
An Empirical Evaluation of the Prospective International SPICE Standard, Journal of Software Process Improvement and Practice, John Wiley, Vol.2, No. 2, 1996, pp. 123-148

H. Günther, D. Rombach, G. Ruhe:
Kontinuierliche Qualitätsverbesserung in der Software-Entwicklung, Wirtschaftsinformatik, 38, 1996, 160-171

C. Lott, D. Rombach:
Repeatable Software Engineering Experiments for Comparing Defect-Detection Techniques, International Journal of Empirical Software Engineering, Vol. 1, No. 3, pp. 241-277. 1996

J. Miller, J. Daly, M. Wood, A. Brooks, M. Roper:
Statistical Power and Its Subcomponents - Missing and Misunderstood Concepts in Software Engineering Empirical Research, Information and Software Technology, vol. 39, No. 4 , pp. 285-295

D. Rombach et al.:
New Institute for Applied Software Engineering, Software Process Newsletter No. 7, Fall 1996, in: Software Engineering Technical Council Newsletter Vol. 14, No. 4, pp.12-14

D. Rombach et al.:
New Institute for Applied Software Engineering, Software Process Journal, 2:2, 1996

Conference Proceedings

M. Baentsch, P. Rösch:
Reviewing two Multimedia Presentation (quasi-) Standards, Proceedings of the International Workshop on Multimedia Software Development, 1996, IEEE Computer Society Press, Berlin Germany, March 1996, pp. 140-149, held together with the 18th International Conference on Software

V. Basili, L. Briand, S. Condon, W. Melo, J. Valett:
Understanding and Predicting the process of software maintenance releases, International Conference on Software Engineering 18, 1996, IEEE Computer Society Press, Berlin, Germany, March 1996

Andreas Birk:
Factors of systematic technology transfer, Proceedings of the Workshop on Technology Transfer, 1996, IEEE Computer Society Press, Berlin Germany, March 1996, held together with the 18th International Conference on Software Engineering

M. Boldt, B. Götze, A. Keller, M. Plessow, P. Rösch:
Werkzeuge zur automatisierten Dokumentation von hierarchischen netzartigen Strukturen, Informatik'96, Softwaretechnik und Standards, GI-Tagung, Klagenfurt, Austria, September 25-27, 1996

L. Briand, C. Differding, D. Rombach:
Practical Guidelines for Measurement-based Process Improvement, Proceedings of the International Software Consulting Network Conference (ISCN'96), Brighton, UK, December 1996

C. Bunse, E. Kamsties:
Integrating SCR Requirements into Cleanroom Software Engineering, Proceedings of the First ISEW Cleanroom Workshop, G. Smith, IEEE Computer Society Press, pp. 41-52, Berlin, Germany, March 1996

C. Bunse, E. Kamsties:
Cleanroom Software Engineering: A Requirements Engineer's View, Proceedings of the Third International Conference on Cleanroom Software Engineering Practices, Q-Labs Inc., College Park Maryland USA, Cleanroom, October 1996

J. M. DeBaud:
Lessons from a Domain-Based Reengineering Effort, Proceedings of the Third Working Conference on Reverse Engineering, pp. 217-226, Monterey, CA, USA, November 1996

J.-M. DeBaud:
An Approach to Address the Problem Space Question in Domain Modeling' Proceedings of the Workshop on Domain Analysis, Conference on Object-oriented Programming Systems, Languages and Applications '96, San Jose, CA, USA, October 6, 1996

J.-M. DeBaud:
Viewing a DSSA in Context: Problems versus Solutions, Proceedings of the 2nd International Software Architecture Workshop (ISAW-2), San Fransisco, CA, USA, October 14-15, 1996

K. El Emam, D. R. Goldenson:
Description and evaluation of the SPICE Phase one trials assessments, Proceedings of the International Software Consulting Network Conference (ISCN'96), Brighton, UK, December 1996

J.-F. Girard:
Reengineering for Maintenance in the year 2000, extended abstract for the 10th European Software Maintenance Workshop, Durham, UK, September 22-25, 1996

C. Gresse, H.D. Rombach, G. Ruhe:
A Practical Approach for Building GQM-Based Measurement - Lessons Learned From Three Industrial Case Studies, Proceedings of the X Brazilian Symposium on Software Engineering, pp. 2-44, Sao Carlos, Brazil, October 1996

B. Heumann, T. Leidig, P. Rösch:
GLASS-Studio: An Open Authoring Environment for Distributed Multimedia Applications, Proceedings of the European Workshop on Interactive Distributed Multimedia Systems and Services, IEEE Computer Society Press, pp. 45-58, Berlin, Germany, March1996

T. Leidig, P. Rösch:
Authoring MHEG Presentations with GLASS-Studio, Proceedings of the International Workshop on Multimedia Software Development, 1996, IEEE Computer Society Press, Berlin, Germany, March 1996, pp. 150-158, held together with the 18th International Conference on Software

F. van Latum, M. Ovio, B. Hoisl, G. Ruhe:
No Improvement Without Feedback: Experiences From Goal-Oriented Measurement at Schlumberger, Proceedings of the Fifth European Workshop on Software Process Technology (EWSPT'96), Nancy, France, October 1996, Lecture Notes in Computer Science, Vol.1149, pp. 67-182

P. Rösch:
User Interaction in a Multi-View Design Environment, Proceedings of the IEEE Symposium on Visual Languages (VL'96), IEEE Computer Society

Press, Boulder Colorado, USA, September 1996

D. Rombach:
Descriptive Software Process Modeling - A first step towards improvement, Seminar, Summerschool of the University of Bari, Bari, Italy, June 03, 1996.

D. Rombach:
Introduction to Measurement, Presentation and Moderation of Measurement Symposium, First European SEPG, Amsterdam, Netherlands, June 24, 1996.

D. Rombach:
Cleanroom Past, Present and Future: Predictions and Prophecies, 3rd International Conference on Cleanroom Software Engineering Practices, Panel presentation, College Park, MD, USA, October 10-11, 1996

D. Rombach:
Position Paper on Experimentation, Measurement Workshop, International Conference on Software Maintenance, Monterey, USA, November 09, 1996

G. Ruhe
Qualitative Analysis of Software Engineering Data Using Rough Sets. Proceedings of the Fourth International Workshop of Rough Sets, Fuzzy Sets and Machine Discovery (RSFD '96), November 1996, Tokyo, Japan, pp 292-299

G. Ruhe:
Rough Set-Based Data Analysis in Goal-Oriented Software Measurement, Proceedings of the Third International Software Metrics Symposium, Berlin, Germany, March 1996, IEEE Computer Society Press, pp. 10-19

G. Ruhe:
Rough Set Based Decision Support in Goal-Oriented Software Measurement, Proceedings of the Workshop on the Advances in Methodology and Software in Decision Support Systems, Laxenburg, Austria, September 1996, pp. 44-46

M. Verlage:
Erfahrungen bei der Formalisierung von Projekthandbüchern, Tagungsband der GI/OCG Jahrestagung Informatik '96,Oldenbourg Verlag München - Wien, September 1996, pp.383-402

Technical Reports

L. Briand, C.Bunse, J. Daly, C. Differding:
An Experimental Comparison of the Maintainability of Object-Oriented and Structured Design Documents, ISERN-Report 96-13, October, 1996

L. Briand, C. Differding, D. Rombach:
Practical Guidelines for Measurement-based Process Improvement, ISERN-Report 96-05, 1996

S. Cinti, A. Fugetta, C. Gresse:
Customized Establishment of Measurement Programs - CEMP, IESE, IESE-Report 01.96/E, July, 1996

C. Differding, D. Rombach:
Kontinuierliche Software-Qualitätsverbesserung in der industriellen Praxis, Internal Report 282/96, University of Kaiserslautern, 1996

C. Differding, B. Hoisl, C. Lott:
GQM Technology Package, Internal Report 281/96, University of Kaiserslautern, 1996

K. El Emam, L. Briand, R. Smith:
Assessor Agreement in Rating SPICE Processes, IESE 1996, September 1996, ISERN-Report 96-09

K. El Elmam, S. Quintin, N. H. Madhavji:
User Participation in the Requirements Engineering Process, REJ, 1996, 1, 1, 4-26

C. Lott:
A Controlled Experiment to Evaluate On-line Process Guidance, University of Kaiserslautern, April 1996

C. Lott:
Measurement-Based Feedback in a Process-Centered Software Engineer-

ing Environment, Internal Report 283/96, University of Kaiserslautern, 1996

S. Morasca, G. Ruhe:
A Comparative Study of Two Techniques for Analyzing Software Measurement Data, ISERN-Report 96-07

J. Reinert, H.-P. Steiert:
Objektorientierte Softwareentwicklung, IESE 1996, December 1996, IESE-Report 03.96/D

M. Verlage:
Erfahrungen bei der Formalisierung von Projekthandbüchern, Fachbereich Informatik Universität Kaiserslautern, 1996, Bericht des Sonderforschungsbereich 501, Nr. 08/96, August 1996

I. Wieczorek:
Aufbau und Durchführung eines Messprogramms bei der Robert Bosch GmbH, Software-Technology-Transfer-Initiative Kaiserslautern, 1996, STTI-96-01-D, University of Kaiserslautern Published in two parts, August 1996

Presentations

L. Briand:
Measuring Software Complexity, ICSE'96, Panel on Software Complexity Measurement, Berlin, Germany, March 1996

C. Bunse:
Reviews und Inspektionen, SoftQuali-Meeting, Heidelberg, Germany, July 11, 1996

Peter Rösch:
Authoring MHEG Presentations with GLASS-Studio, International Workshop on Multimedia Software Development, (workshop at ICSE'96), Berlin, Germany, March 25-26, 1996

D. Rombach:
Software-Entwicklungs-Know-how. Ein entscheidender Faktor für die industrielle Wettbewerbsfähigkeit. Kolloquium, Institut für Techno- und Wirtschaftsmathematik e.V. (ITWM), Trier, Germany, September 25, 1996

G. Ruhe:
How to Organize a Repository of Best Practices in Software Engineering: A Study for the European Software Institute, STTI-94-01-E

G. Ruhe:
Kosten-Vorhersagemodelle als Bestandteil unternehmensweiter Erfahrungsdatenbanken, Kolloquium, Fachbereich Informatik, Universität Kaiserslautern, Germany, Juni 1996

Doctoral Theses

C. M. Lott:
Measurement-Based Feedback in a Process-Centered Software Engineering Environment,
Computer Science Department,
University of Maryland,
College Park, MD, USA,
Advisor: Prof. Dr. Dieter Rombach
May 1996

Alfred Bröckers:
Modellbasierte Analyse von Software-Projektrisiken,
Department of Computer Science,
University of Kaiserslautern,
Advisor: Prof. Dr. Dieter Rombach
November 1996

Diploma Theses
(Diplomarbeiten)

Torsten Armbrecht:
Analyse von GQM-Plänen auf Grundlage von Rough Sets,
Department of Computer Science,
University of Kaiserslautern,
Supervisor: Prof. Dr. Dieter Rombach,
Dr. Günther Ruhe,
January 1996

F. Gieseke:
Entwicklungsunterstützung von JAVA-basierten Multimedia-Präsentationen,
Department of Computer Science,
University of Kaiserslautern,
Supervisor: Peter Rösch,
October 1996

M. Jahn:
Ein Werkzeug zum Einsatz von Metriken bei der Bewertung von objektorientierten Softwareentwürfen,
Department of Computer Science,
University of Kaiserslautern,
Supervisor: Peter Rösch,
September 1996

B. Völker:
Entwicklung eines Kernmodells zur Unterstützung von Multi-View-Editoren, Department of Computer Science, University of Kaiserslautern,
Supervisor: Peter Rösch,
October 1996

Project Theses
(Projektarbeiten)

Markus Hoffmann:
Plattformunabhängige Werkzeugunterstützung der Planungsphase des GQM-Prozesses,
Department of Computer Science,
University of Kaiserslautern,
Supervisor: Andreas Birk,
September 1996

D. Klein:
Ein Werkzeug zur Source Code Transformation,
Department of Computer Science,
University of Kaiserslautern,
Supervisor: Christian Bunse,
February 1996

Awards

External

J.M. DeBaud:
Best Paper Award at the 3rd Working Conference on Reverse Engineering in Monterey, CA, USA, November 9, 1996

L. Briand:
IEEE Award for an outstanding contribution to the organization of ICSM'96, Monterey, CA, USA, November 1996

D. Rombach:
IEEE Senior Membership Award, December 1996

D. Rombach:
IEEE Certificate of Appreciation for having served on the IEEE Software Magazine Editorial Board 1992-1996, November 1996

D. Rombach:
ACM Recognition of Service Award 1996

Internal

C. Gresse:
The Fraunhofer IESE 1996 Award for Research Excellence

F. Huber:
The Fraunhofer IESE 1996 Award for Technical Support Excellence

O. Laitenberger:
The Fraunhofer IESE 1996 Award for Research Excellence

D. Pfahl:
The Fraunhofer IESE 1996 Award for Project Excellence

I. Wieczorek:
The Fraunhofer IESE 1996 Award for Project Excellence

Events, Facts, and Photos

Chronicle (highlights)

January
Foundation of the Fraunhofer Institute for Experimental Software Engineering, January 1

February
Opening ceremony of the Fraunhofer IESE, February 14

March
18th International Conference on Software Engineering, ICSE 18, Berlin, March 25

Presentation of the Fraunhofer IESE at the International Conference on Software Engineering ICSE 18, Berlin, March

May
Press conference (together with Markant Südwest Software- und Dienstleistungs GmbH), Kaiserslautern, May 9
1) Presentation of the newly-founded Markant Südwest Software- und Dienstleistungs GmbH
2) Information about the collaboration between Markant Südwest Software- und Dienstleistungs GmbH and Fraunhofer IESE

August
Signing of the collaboration agreement between CAESAR and Fraunhofer IESE, August 27

September
Talk by Prof. Dr. David Lorge Parnas, "Mathematical Description and Specification of Software", September 2

Talk by Prof. Dr. Thomas Seewaldt, "Psychology of Social Systems and the Professionalization of Software Teams", September 19

First meeting of the IESE Advisory Board, Kaiserslautern, September 6

October
Talk by Prof. Dr. Gregor Bochmann, "Systematic test development from system specifications including real-time aspects", October 8

Talk by Reidar Conradl, "SPIQ: A Revised Agenda for Software Process Support", October 11
Presentation by Robert Susmaga, "Combination of Fuzzy Set Theory and Rough Set Theory to Analysis of Data Containin Uncertainties", October 17

November
Talk by Prof. Dr. Ross Jeffery, "How Can We Reduce Cycle Time and Cost? - Opportunities in Software", November 29

December
Talk by Dr. Colin Atkinson, "Towards the Integration of Object-Oriented Notations, Architectures and Processes", December 9

Election of the IESE deputies for the Scientific Advisory Board of the Fraunhofer Gesellschaft, Kaiserslautern, December 10; elected: Dr. Frank Bomarius, Dr. Peter Rösch

Opening of the Fraunhofer IESE, February 14, 1996

Events in Pictures

Prof. Dr. Ross Jeffery, giving a talk about:
"How Can We Reduce Cycle Time and Cost? - Opportunities in Software Inspections/Technical Reviews"
November 29, 1996

Prof. Dr. David Lorge Parnas, giving a talk about:
"Mathematical Description and Specification of Software"
September 2, 1996

Signing the collaboration agreement between CAESAR and Fraunhofer IESE
from left to right: Prof. Dr. Rombach, Prof. Dr. R. A. Layton, Prof. Dr. Jeffery.
August 27, 1996

Active partnership - even on the soccer field: Soccer team of Q-Labs (in red shirts) and of Fraunhofer IESE after a match.
May 11, 1996

Prof. Dr. Thomas Seewaldt (FH Mannheim, Germany), giving a talk about:
"Psychology of Social Systems and the Professionalization of Software Teams"
September 19, 1996

Hand shake on the occasion of the foundation of the Markant Südwest Software- und Dienstleistungs GmbH, Kaiserslautern, May 9, 1996
From left to right: Franz Mayer, Chairman, Markant Südwest AG; Florian Bernauer, Executive Manager, Markant Südwest Software- und Dienstleistungs GmbH; Prof. Dr. Dieter Rombach, Fraunhofer IESE; Wolfgang Jacob, Executive Manager, Markant Südwest Software- und Dienstleistungs GmbH; Gerhard Piontek, Mayor of Kaiserslautern.

Prof. Dr. Dieter Rombach, addressing a heartfelt welcome to the participants of the 18th International Conference on Software Engineering, ICSE 18, Berlin, March 25, 1996

Press conference on the occasion of ICSE 18
The panel from left to right: Prof. Ernst Denert, Prof. Dr. Marvin Zelkowitz, Prof. Dr. Dieter Rombach, Stephan Drooff, Prof. Dr. Stefan Jähnichen, Prof. Mario Barbacci

Media Coverage of the Fraunhofer IESE

Reports and articles about the opening of the Fraunhofer IESE and other IESE-related themes have been published in the following media:

- Allgemeine Zeitung, 02-15-1996
- Blick durch die Wirtschaft, 01-05-1996
- Computer Zeitung, 01-18-1996, 03-14-1996
- Deutschlandfunk, Forschung aktuell, 03-30-1996
- Der Fraunhofer, No. 2/1996
- Heizung, Lüftung - Klima Haustechnik, No. 4/1996
- Industrie Anzeiger, 03-11-1996
- Mainzer-Rhein-Zeitung, 02-15-1996, 02-16-1996
- Maschinen Markt MM, No. 10/1996
- Pfälzischer Merkur, 02-15-1996
- Rheinpfalz,02-13-1996, 02-15-1996, 05-10-1996
- Software Process Improvement and Practice, 10-18-1996
- Software Process Newsletter, No. 7, Fall 1996
- Standort Chemie, No.4/1996
- Südwestfunk 1, Rheinland-Pfalz aktuell, 02-14-1996
- The Australian, 08-28-1996
- Trierischer Volksfreund, 02-15-1996
- Uni-Spectrum Kaiserslautern, No. 2/1996, 3/1996, 4/1996
- VDI-Nachrichten, 03-01-1996
- Wirtschaftsmagazin Pfalz, No. 4/1996
- Wissenschaft, Wirtschaft, Politik, 02-21-1996

Humor

When top managers of the Fraunhofer Gesellschaft come together - on the occasion of the press conference during the opening of the Fraunhofer IESE, Kaiserslautern, February 14, 1996

Really, my dear Rombach: Don't you think your institute's name turned out to be just a little bit too long?

Not at all, my dear Warnecke. My GQM experts have measured it and found out that it consists only of one line approximately this long.

Compared to the million lines of code that we usually work with, we found this one to be pleasantly short.

The Fraunhofer-Gesellschaft

The Research Organization

The Fraunhofer-Gesellschaft is the leading organization of applied research in Germany. It operates 47 research institutes with about 8,800 employees, about half of them scientists and engineers. The research volume for 1996 will amount to over one billion DM; more than two-thirds of this amount is earned through contracts from industry and the public sector (>50% of the industrial earnings come from small- and medium-sized enterprises).

International activities are increasingly important. Apart from the collaboration with numerous companies and research establishments within Europe, the Fraunhofer-Gesellschaft operates resource centers and research units in the United States and Asia. The Fraunhofer Management Gesellschaft (FhM) was founded as a subsidiary company in 1990.

The appellation, Fraunhofer-Gesellschaft, was chosen in reference to the researcher, inventor, and entrepreneur Joseph von Fraunhofer (1787 - 1826), who won widespread acclaim for his scientific and commercial achievements.

The Research Fields of the Fraunhofer-Gesellschaft

Eight fields form the core of Fraunhofer research:

- Materials and Components
- Production Technology
- Information and Communication
- Microelectronics and Microsystems
- Sensor Systems, Testing Technologies
- Process Engineering
- Energy, Environment, Health
- Technical and Economic Studies

Apart from research services, certified test beds and other facilities can also be provided.

Advantages of Contract Research with the Fraunhofer-Gesellschaft

More than 2,600 experts are available for the development of complete systems.

All developments are based on profitability considerations.

The Fraunhofer-Gesellschaft collaborates with various renowned companies whose research contracts have resulted in successful products.

Modern laboratory equipment and scientific aids such as project management and internationally-linked communications systems enhance the quality of the research work.

Detailed project reports, instructions for use, staff training, and complete introduction strategies for new technologies round off the contract research services.

Reliability, continuity, and service of a large organization are available to all companies.

Collaboration with the Fraunhofer-Gesellschaft

Contract research with the Fraunhofer-Gesellschaft has advantages for all companies. Orders come from all branches of industry and from companies of all sizes. The institutes' facilities are particularly recommended for small businesses who can take advantage of Fraunhofer research when their own capacities are not sufficient to develop on their own the technical innovations necessary to stay competitive.

Executive Board

Prof. Hans-Jürgen Warnecke, President
Dr. Dirk-Meints Polter, Personnel
Dr. Hans-Ulrich Wiese, Legal Department

Accounting Department Address

Fraunhofer-Gesellschaft
Leonrodstraße 54
D-80636 München, Germany

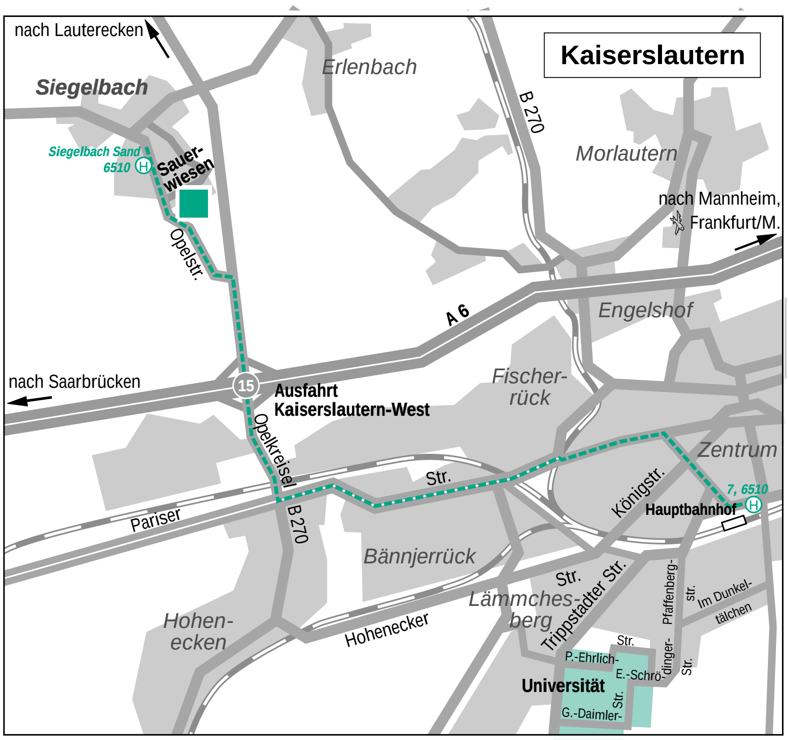
Tel	+49 89 1205-577, -544
Fax	+49 89 1205-317
email:	info@zv.fhg.de
WWW:	http://www.fhg.de

Fraunhofer locations

Fraunhofer IESE Contact

Fraunhofer Institute for
Experimental Software Engineering
Sauerwiesen 6
D-67661 Kaiserslautern
Tel: +49(0)6301/707-100
Fax: +49(0)6301/707-200
E-Mail: info@iese.fhg.de

Our web server offers up-to-date information about the institute. We invite you to visit our web site at: http://www.iese.fhg.de



How to reach us:

- by car
  coming from the west (Saarbrücken) or the east (Mannheim) on highway (Autobahn) A6. Take the exit"Kaiserslautern-West“ and follow the signs that read "Lauterecken". About 500 m after exiting the highway, turn left to "Siegelbach". Follow the road leading through a forest. Right after entering "Siegelbach" you turn right at the first junction into the street "Sauerwiesen". After about 100 m you find IESE on your right-hand side.

- by train
  from Kaiserslautern railway station either by taxi (ca. 8 km) or by bus (line RSW 6510, departing from bus stop A/2 at railway station, destination: Siegelbach) to Siegelbach; the stop "Siegelbach Sand" is about 100 m from the institute

- by airplane
  Airport Frankfurt/Main, either by train (about 2 hours) or by car (about 1.5 hours)

Fraunhofer IESE Contact

Dial Phone No. +49(0)6301/707- ...

100	Prof. Dr. Dieter Rombach Director rombach@iese.fhg.de
121	Dr. Günther Ruhe Deputy Director Department Head ZDÖP (Central Services and Publicly-funded Projects) ruhe@iese.fhg.de
121	Dr. Frank Bomarius Department Head IQVP (Industrial Quality Improvement Projects) bomarius@iese.fhg.de
251	Dr. Lionel Briand Department Head QPE (Quality and Process Engineering) briand@iese.fhg.de
251	Dr. Jean-Marc DeBaud Department Head ISE (Innovative Software Engineering) debaud@iese.fhg.de
262	Dr. Klaus Hörmann Center for Small and Medium Enterprises (SME) hoerman@iese.fhg.de
122	Joachim Müller-Klink Public Relations mkl@iese.fhg.de