



# ANNUAL REPORT 2015

## Contact

[office@swc.rwth-aachen.de](mailto:office@swc.rwth-aachen.de)

[www.swc.rwth-aachen.de](http://www.swc.rwth-aachen.de)

+49-241-80-21331

Ahornstr. 55

52074 Aachen, Germany

## Staff

- **Faculty**

Univ.-Prof. Dr. rer. nat. Horst Lichter  
lichter@swc.rwth-aachen.de  
www.swc.rwth-aachen.de



- **Secretary**

Marion Zinner  
Phone: +49 241 80 21 330  
Fax: +49 241 80 22 352  
Email: office@swc.rwth-aachen.de



- **Research Assistants**

Dipl.-Inform. Matthias Vianden  
(third-party funds position)

Ana Nicolaescu, M.Sc.  
(third-party funds position)

Muhammad Firdaus Harun, M.Sc.  
(Malaysia Government scholarship)

Andrej Dyck, M.Sc.

Dipl.-Inform. Andreas Steffens  
(third-party funds position)

Konrad Fögen, M.Sc.  
(third-party funds position)

Simon Hacks, M.Sc.  
(third-party funds position)



- **External Researchers**

Dipl.-Inform. Andreas Ganser  
Tanya Sattaya-aphitan, M.Eng.  
(TOT scholarship)



- **Student Researchers**

Peter Alexander  
Tien Dung Le  
Artjom Göringer  
Jan Simon Döring  
Arthur Otto  
Stefan Dollase  
Yalcin Bogazci



## Overview

Our research focuses on the development of new methods, tools, and techniques in the broad area of software construction. Since we always aim to develop results that are applicable under industrial software development conditions, most of our research projects are performed in close cooperation with industrial partners. Currently, we are doing research in the following areas (see Projects for more details):

- *Reusing Domain Engineered Artifacts for Code Generation*
- *Metric-based Project and Process Management*
- *Cost Benefit based Technical Debt Management*
- *Model-based Software Architecture Evolution and Evaluation*
- *Regression Test Optimization*
- *DevOps-aware Software Engineering*
- *Architecture Roundtrip Management*
- *Test Automation for heterogeneous Application Landscapes*

Since appropriate tools are often door openers to transfer research ideas to practice, we are developing dedicated tool support for those areas. Currently, we offer:

- HERMES (Harvest, Evolve, and Reuse Models Easily and Seamlessly)
- EMI (Enterprise Measurement Infrastructure) and SCREEN
- ARAMIS (Architecture Analysis and Monitoring Infrastructure)

Last year, we organized the 2<sup>nd</sup> International Workshop on Quantitative Approaches to Software Quality which was held in conjunction with APSEC 2014 in Jeju, Korea. Currently we are organizing the follow-up workshop co-located with APSEC 2015 in New Delhi, India.

In December 2014, Simona Jeners successfully passed her Ph.D. defense on *Model-supported Process Adoption and Assessment in the Context of Multiple Practice Repositories*. Matthias Vianden completed his research project on Metric-based Project and Process Management; his Ph.D. defense will be in December 2015.

Furthermore, we are proud to start four new joined industry cooperation projects with Generali, ITERGO, IVU, and Kisters (see Projects for more details). Last but not least, we are happy to welcome Konrad Fögen and Simon Hacks in our team. They are doing their research projects in the context of the new Generali and ITERGO project respectively.

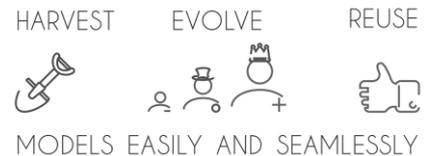
# Research Projects

## Reusing Domain Engineered Artifacts for Code Generation

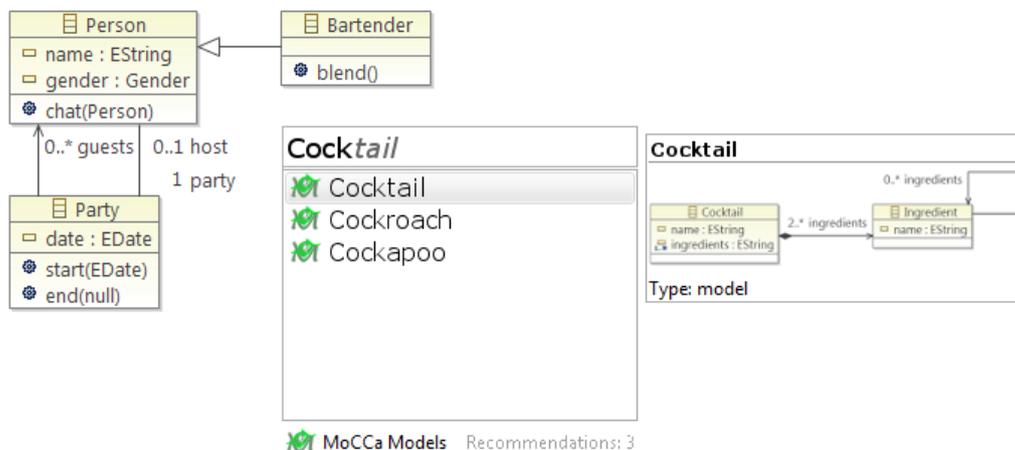
A. Ganser, H. Lichter

Model driven architecture (MDA), and model driven engineering (MDE) are promising approaches to increase reuse and to reduce development time and effort. Both comprise of several methods which include domain specific modeling (DSM). These methods brings about figures which map objects under consideration to models. Among these models are class diagrams as know from UML. They are called domain models in these contexts.

## HERMES



Both approaches take these domain models as inputs for code generation, but only MDE includes reuse in DSM. Yet, this reuse remains rather rudimentary. Taking a closer look at model repositories one might suppose that these repositories are meant to store models so they can be reused rather easily in different projects. But the goals for these repositories are totally different! All the available repositories (by and large) only consider versioning, migration, transformation, conflict detection, merging and querying. This means, models are not related to each other, there is barely a description of models, no examples are present how the models could be used or no interfaces are defined which point to the most important aspects that could help reusing a particular model.



The goal of this research project is to bolster model reuse by providing mechanism to harvest, evolve, and reuse models. Therefore approaches for gleaning reusable artifacts into a model library, evolving them, and producing recommendations are under research. Therefore, models should not be treated as in an isolated world, but related to each other, knowing not only that these models worked together but even how they did. These relationships cross borders and overcome the usual reuse obstacles and unleash the full power of previously modeled knowledge.

For more information please visit: <http://hermes.modelrecommenders.org>

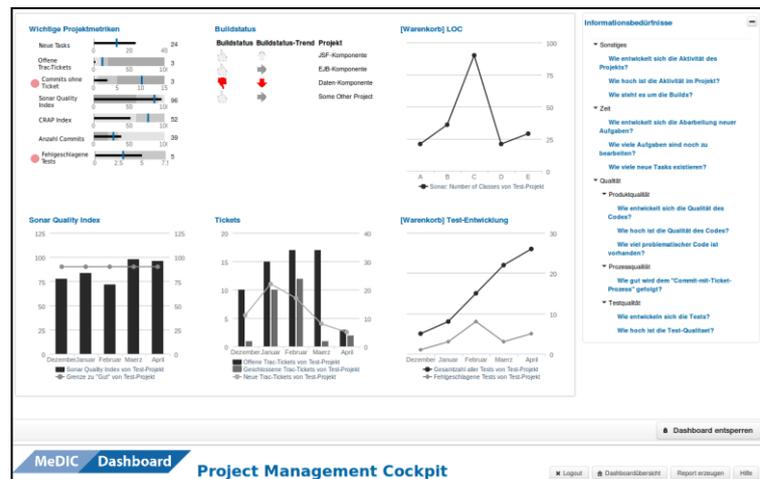
## Metric-based Project and Process Management

M. Vianden, H. Lichter

External cooperation: Generali Informatik Services, Aachen

Projects management greatly benefits by the application of metrics. However, research shows that it is demanding to find the right metrics; 58% of all project managers and 50% of all senior managers find it difficult to collect, analyze, and use the right metrics. On the one hand, metric frameworks like GQM help to derive metrics from abstract goals for the project. On the other hand, defining measures just for one project (in a multi project organization with a lot of similar projects) is costly and ineffective.

Although considerable research has been devoted to the modeling of metrics and metric frameworks, rather less attention has been paid to investigating how the results of this research can lead to a sound concept for metric reuse. Therefore, the aim of this project was to develop a systematic approach to define metrics and implement the metrics based on a dedicated metric infrastructure and a metric system development process. This enables a company-wide systematic metric reuse.



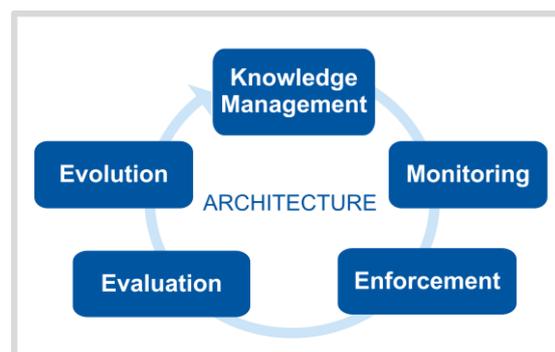
## Model-based Software Architecture Evolution and Evaluation

A. Nicolaescu, H. Lichter

External cooperation: Generali Informatik Services, Aachen

The architecture of software systems directly influences crucial quality attributes and therefore should be considered whenever important decisions regarding their evolution must be taken. However, up-to-date descriptions that correctly reflect the system's architecture rarely exist. Architecture descriptions are usually elaborated at the beginning of a software project. After the initial version of the system has been constructed, the system tends to evolve independently from its architecture description. Changes to the system are rarely documented properly and originally imposed rules are gradually violated.

To help architects manage this situation, we have developed ARAMIS, – the Architecture



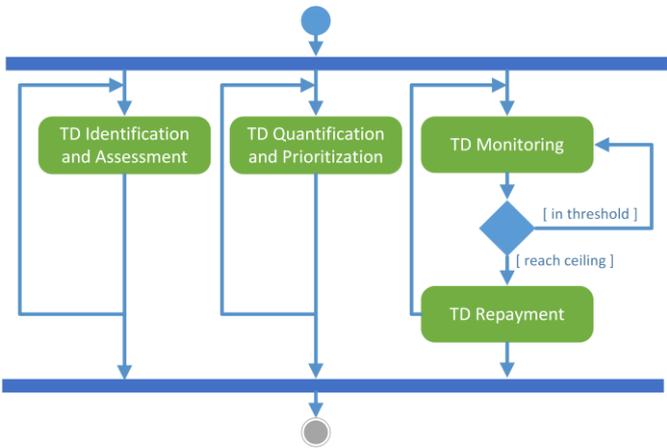
Analysis and Monitoring Infrastructure. Its main goal is to support the understanding and architectural evaluation of software systems. It supports these goals by monitoring and analyzing the run-time of the considered system, pursuing to answer questions such as: “how do the architecture units interact with each other upon performing a certain scenario (e.g., running a test-case, interacting with the graphical user interface, etc.)?”, “which are the architecture units that need to be redesigned?”, “which are the various hot spots of the system (e.g., in terms of received calls, outgoing calls, caused violations, etc.)?”, “are there violations against the architecture description?”, etc.

To achieve this, ARAMIS validates the communication inside a software system during the execution of some scenarios of interest. The resulted calls are mapped on architecture units and checked against predefined architecture rules. The results are then visualized to support the architect to discover the architectural weaknesses of the analyzed system.

We currently work on extending our solution to increase the level of support given to the architects by, e.g., offering them the possibility to reuse previously created architecture descriptions (instead of redefining these using an ARAMIS-specific meta-model) and by enabling the application of standard and self-defined views on the extracted behavior in order to facilitate its understanding by considering only relevant details.

**Cost Benefit based Technical Debt Management**  
*M. Firdaus Harun, H. Lichter*

Implementing a payback strategy to reduce the technical debt (TD) is a must for every software organization, as TD exists in most software systems. If we do not cautiously manage the debt or have no strategy to pay it back, the system finally may go to the "bankruptcy" phase, i.e., the software is unmaintainable and the maintenance cost will increase continuously. In general, refactoring is one of the strategies to pay it back. Usually, project managers are always juggling on the decision to either implement new features or to make improvements in a release cycle as it is always complicated to decide, which refactoring task should be done first or could be postponed.

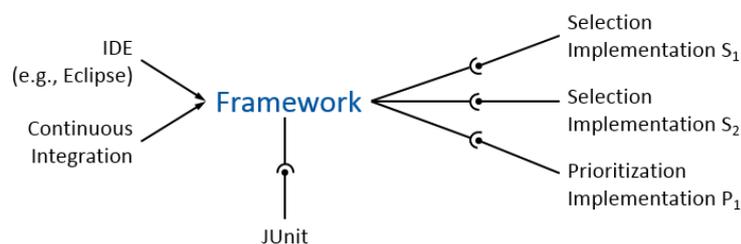


To this end we propose an approach called Cost-Benefit based Technical Debt Management (CoBeTDM). It aims to help business and technical people to decide which refactoring/maintenance should be implemented in a short-term or long-term. At first, TD items are identified on detected code and architecture smells using specified metrics. Then the identified TD items are quantified and prioritized applying a Cost-Benefit analysis taking also into consideration the risks associated to respective refactoring tasks. Based on the Cost-Benefit analysis Return on Investment values can be determined which can be used to decide when to implement which refactoring in order to pay back the TD step by step.

## Regression Test Optimization

*Andrej Dyck, H. Lichter*

Software regresses when existing functionality stops working upon the change of the program. Thus, the importance of automatic regression testing is increasing; this is especially true for fast-evolving systems and continuous delivery done right. To stress functional behavior, the regression test suite and its execution time is, in general, large. Moreover, by definition, regression test suites are executed recurrently, and thus, the number of test cases greatly influences the execution time. However, a change has only a partial impact on the system, so we can choose to execute only relevant test cases (test selection). Similarly, we can



permanently remove test cases that are “irrelevant” (test minimization). Last but not least, the order of the test cases can reveal regression errors “faster” (test prioritization).

In research, several optimization techniques as proposed.

However, those strategies often exist only on paper, don’t scale, and tools are very rare to non-existent. We are working on a framework for Java and JUnit that allows an easy integration into existing projects and enables researchers to implement new optimization strategies. Moreover, a concept for pointing out “irrelevant” tests and automated evaluation for strategies has to be developed. Further, we want to identify prerequisites and requirements for projects to be suitable for test selection and prioritization techniques.

## DevOps-aware Software Engineering

*Andreas Steffens, H. Lichter*

*External cooperation: KISTERS AG and IVU AG, Aachen*

Each software system relies on infrastructure elements to be operated correctly. Caused by the growing complexity and distributed manner of modern software systems the management of infrastructure gets more and more important. In parallel, the frequent delivery and stable operation of software gain more attention. A tight integration of methods from software engineering and software operation can achieve this goal.

Faced with these challenges new approaches and methods are proposed. Continuous Integration and Continuous Delivery focus on the automation of software building and testing in a deployment pipeline.

Fueled by this approach infrastructure automation is integrated in the software development process. Infrastructure is described and handled in a declarative manner like normal source code. Common known software engineering methods were transferred to the infrastructure

and operation domain. This all leads to the holistic approach of DevOps or Continuous Software Engineering (CSE).

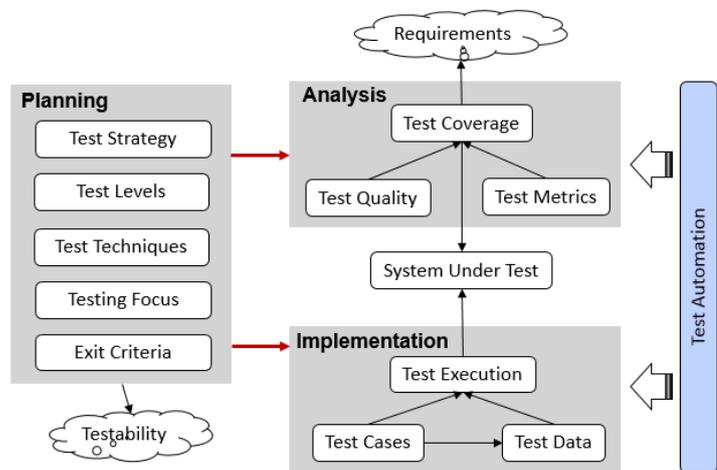
This project aims to identify and evaluate a systematic approach of introducing and adapting methods from DevOps or CSE into the software development processes of two mid-sized software companies.

## Test Automation for Heterogeneous Application Landscapes

*Konrad Fögen, H. Lichter*

*External cooperation: Generali Informatik Services, Aachen*

Heterogeneous application landscapes distinguish themselves by high complexity and the usage of many different technologies. In such an environment, it is of utmost importance to frequently integrate the heterogeneous parts and to test them thoroughly. However, frequent and thorough testing activities costs time and reduces the overall efficiency of the software development. A high degree of test automation can significantly improve the efficiency of the test process. Although, that does not necessarily incorporate the effectiveness of the test process, as it requires a systematic procedure in preceding test-related activities.



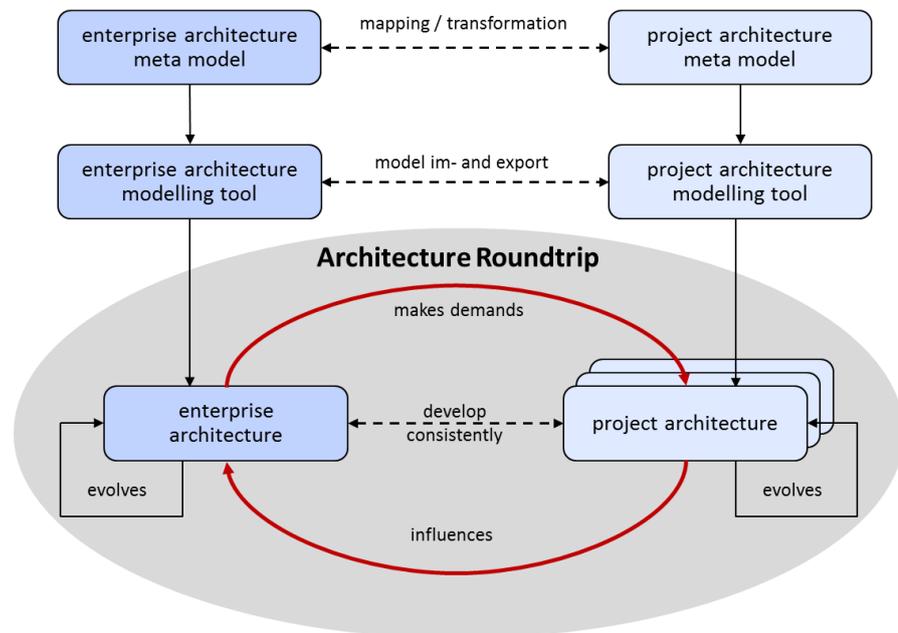
This research project aims to improve the test efficiency as well as effectiveness by increasing the degree of test automation among all test levels and all test activities. Therefore, we are researching approaches to establish extensive test automation in complex industrial settings. This includes the identification of impediments and weaknesses that currently impede the automation of test activities, the development and adaption of test automation frameworks as well as of concepts to automatically assess the quality of tests.

## Architecture Roundtrip-Management

Simon Hacks, H. Lichter

External cooperation: ITERGO Informationstechnologie GmbH, Düsseldorf

Within Enterprise Architecture Management (EAM) guidelines and models are defined to be used or refined in projects. Although the developed project architectures should be conformant to the EAM models, unfortunately the project architectures often differ from the planned enterprise architecture. This changes have to be transferred from the project architectures to the as-is enterprise architecture orchestrated by a quality management process. Additionally it should be assessed how these changes affect the planned enterprise architecture. In this project, an approach should be developed aiming at a continuous and systematic alignment of the project architectures to the enterprise architecture and vice versa. To achieve this aim the integration of the tools used by our cooperation partner has to be promoted. This includes the adjustment of related processes, exchange of information between the tools and the delivery of all needed information to the EAM customers.



Based on the consistent architecture the current state can be assessed and alternative evolutionary scenarios can be created and evaluated. To realize this an architecture analysis and rating method should be developed. This implies the identification and measurement of EAM related KPIs.

## Special Events

### 2<sup>nd</sup> International Workshop on Quantitative Approaches to Software Quality (QuASoQ) Jeju, Korea, December 1<sup>st</sup>, 2014

Collocated with the 21<sup>th</sup> Asia-Pacific Software Engineering Conference (APSEC 2014), we organized the 2<sup>nd</sup> Int. Workshop “Quantitative Approaches to Software Quality” (QuASoQ). This workshop aims at gathering together researchers and practitioners to discuss experiences in the application of state of the art approaches to measure, assess and evaluate the quality of both software systems and software development processes. Quantitative approaches are important means for software development organizations to assess their current level of quality and to perform goal-oriented improvement initiatives.



The goals of the workshop are to exchange experiences, present new and promising approaches and discuss how to set up, organize, and maintain quantitative approaches to software quality.



The program committee selected the papers based on their quality and novelty, and finally accepted seven papers covering several topics.

In order to foster a lively discussion, we deviated from the common “authors present, audience asks” scheme. Instead, every author presented the paper of another participant, while the original author served as discussant. The new format, although seen a bit skeptical by the participants at the

beginning, really helped to create intensive discussions on the presented topics.

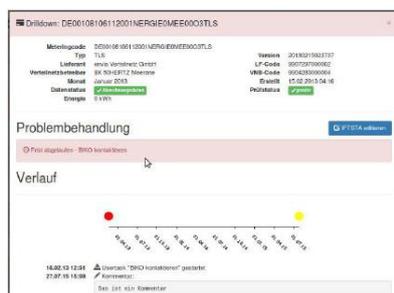
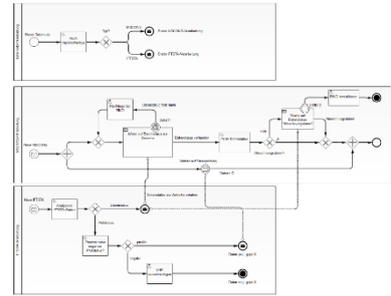
The workshop proceedings are published in volume 2 of APSEC 2014 proceedings at IEEE Digital Library.

Currently we are organizing the 3<sup>rd</sup> International Workshop “Quantitative Approaches to Software Quality” (QuASoQ) that will be held in December 2015 in conjunction with 22<sup>nd</sup> Asia-Pacific Software Engineering Conference in New Delhi, India.

## Tackling a real-world problem in a teaching lab

Due to the good experience in 2014, we decided to involve an industrial cooperation partner in a teaching activity once more. The industry part was represented by KISTERS AG; a long term cooperation partner of our group.

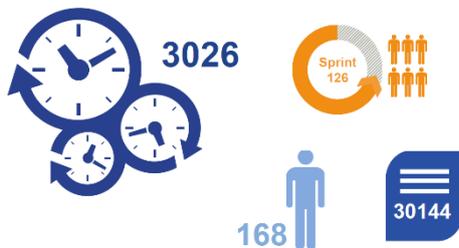
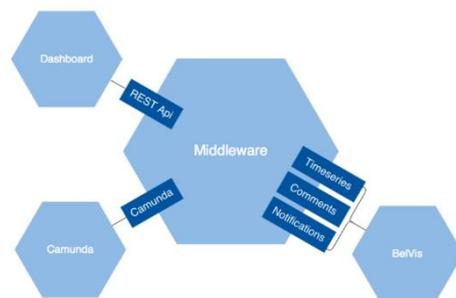
KISTERS offers a broad portfolio of software solutions for companies in the energy sector. The problem to tackle as suggested by KISTERS was to monitor the accounting of energy deliveries of participating companies. The market in Germany is deregulated and a complex process is implemented to ensure the correct management and accounting. Within this process, a large amount of messages (several thousand a day) between the participants are exchanged.



The task for our students was to create a software dashboard to monitor the status of current communication processes. The process itself was implemented using the BPM engine Camunda provided by KISTERS. Additionally, a simulation using historical communication and event data provided by KISTERS had to be implemented. The resulting system consisted of three major components around the BPM engine:

- Dashboard: A web frontend
- Middleware: A service integrating process and historic data
- Simulator: A service using history messages to feed the BPM engine with a year of data in 15 minutes

The lab was organized as an agile software project supported by JIRA-Agile and other Atlassian tools. The students formed three teams of six team members each. Every team individually came up with different solutions for the proposed problems after facing the initial problem of getting familiar with the different implementation technologies consisting of BPM-Engine Camunda for process execution, NodeJS for a scalable middleware, the Java Spring framework, and AngularJS for the Dashboard component.



During the lab, KISTERS provided a project manager who participated in discussions and approved certain milestones, as well as provided access to real data comprising messages from real customers over a period of one year. At the end of the project, each student team presented a working prototype and important insights into the problem domain in a small event hosted by KISTERS. Many of KISTERS' developers and managers were interested and thrilled with the results.

The results and demos were reused by KISTERS in an internal conference to transfer insights and acquired knowledge to their developers and customers.

## Other Activities

- Member of the international program committee, 7<sup>th</sup> Asian Conference on Intelligent Information Systems and Database Systems, Bali, Indonesia, March 23-25, 2015, *H. Lichter*
- Organization and chair of the international program committee, 2<sup>nd</sup> International Workshop on Quantitative Approaches to Software Quality, Jeju, Korea, December 1, 2014, *H. Lichter, M. Vianden*
- Member of the international program committee, 30<sup>th</sup> Annual ACM Symposium on Applied Computing, Software Engineering Track, Salamanca, Spain, April 13-17, 2015, *H. Lichter*
- Member of the international program committee, IEEE International Conference on Software Quality, Reliability & Security (QRS), Vancouver, Canada, August 3-5, 2015, *H. Lichter*
- Member of the program committee, Software Engineering im Unterricht der Hochschulen (SEUH), Dresden, Germany, February 26-27, 2015, *H. Lichter*
- Member of the program committee, Doktorandensymposium der SE 2015, Dresden, Germany, March 18, 2015, *H. Lichter*
- Member of the program committee, Track Software Engineering Ideen der SE 2015, Dresden, Germany, March 19, 2015, *H. Lichter*
- Member of the working group "Scientific software engineering, software eco-system and programmability" of the European Exascale Software Initiative 2, *H. Lichter*
- Reviewer for dpunkt-Verlag Heidelberg and computing reviews, *H. Lichter*
- Organization of the Computer Science Department's mentors program, *H. Lichter*
- Member of the Computer Science Department's committee for Lehre and Service-Lehre, *H. Lichter*
- Member of the examination board of Computer Science, *H. Lichter*
- Member of workgroup "Zusammenarbeit Hochschule und Industrie", GFFT, Gesellschaft zur Förderung des Forschungstransfers, *H. Lichter*
- Member of the International Networking in Science & Technology (INSTec) Advisory Committee, *H. Lichter*
- Lecturer for the "Kara, der programmierbare Marienkäfer" course at Helle Köpfe in der Informatik 2015, *H. Lichter*
- Lecturer at Thai German Graduate School of Engineering (TGGS), course „Software Engineering“, Bangkok, Thailand, *H. Lichter*
- Member of the appointment commission „Applied Software Technique“, University Hamburg, *H. Lichter*
- Reviewer of the Bachelor and Master program „Softwaretechnik“, University Stuttgart, *H. Lichter*
- Organization of a joint workshop on „Digital Lifestyle“ in co-operation with Universiti Teknologi Malaysia, Johor Bahru, December 15-16, 2015, *H. Lichter, A. Steffens*

- Organization of TGGS leadership meeting at the Department of Computer Science, April 27, 2015 , *H. Lichter*
- Organization of the universal and specialized Preparatory Courses in Computer Science 2015, *H. Lichter, A. Dyck*

## Talks and Publications

### Talks

A. Nicolaescu: *The ARAMIS Workbench for Monitoring, Analysis and Visualization of Architectures Based on Run-time Interactions*. SAeroCon, Cavtat/Dubrovnik, Croatia, September 8, 2015.

A. Nicolaescu: *Evolution of Object Oriented Coupling Metrics: A Sampling of 25 Years of Research*. SAM, Florence, Italy, 16 May 2015.

A. Dyck: *Towards Definitions for Release Engineering and DevOps*. RELENG, Florence, Italy, 19 May 2015.

M. Vianden: *Experience on a Microservice-based Reference Architecture for Measurement Systems*. APSEC, Jeju, Korea, December 2, 2014.

M. Vianden: *Lessons Learned on Systematic Metric System Development at a large IT Service Provider*. QuASoC, Jeju, Korea, December 1, 2014.

A. Nicolaescu: *Towards an Architecture Quality Index for the Behavior of Software Systems*. QuASoC, Jeju, South Korea, December 1, 2014.

A. Nicolaescu: *Run-time Monitoring-based Evaluation and Communication Integrity Validation of Software Architectures*. APSEC, Jeju, Korea, December 2, 2014.

H. Lichter: *Introduction to 2<sup>nd</sup> QuASoQ workshop*, Jeju, Korea, December 1, 2014.

H. Lichter: *Systematic Metric Management – Challenges and Solutions*. Mahidol University, Salaya Campus, Bangkok, November 28, 2014.

S. Jeners: *Model-supported Process Adoption and Assessment in the Context of Multiple Practice Repositories*, Doctoral Seminar, December 17, 2014

### Publications

A. Nicolaescu, H. Lichter, A. Göringer, P. Alexander, D. Le (2015): *The ARAMIS Workbench for Monitoring, Analysis and Visualization of Architectures Based on Run-time Interactions*. In SAeroCon, 2<sup>nd</sup> Workshop on Software Architecture Erosion and Architectural Consistency, co-located with the 9th European Conference on Software Architecture (ECSA 2015), Cavtat/Dubrovnik, Croatia, September 8, 2015, ACM, New York, NY, USA, 57:1–57:7.

P. Suwannawiwat, C. Netramai, M. Utech, H. Schlebusch, H. Lichter (2015): *Unifying Hydrological Time Series Data for a Global Water Portal*. In 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Hua Hin, Thailand, June 24-27, 1-5.

S. Jeners (2015): *Model-supported Process Adoption and Assessment in the Context of Multiple Practice Repositories*. Shaker, Aachen.

A. Nicolaescu, H. Lichter, Y. Xu (2015): *Evolution of Object Oriented Coupling Metrics: A Sampling of 25 Years of Research*. In 2nd International Workshop on Software Architecture and Metrics associated with the 37th International Conference on Software Engineering (ICSE 2015), 16 May 2015, Florence, Italy, 48-54.

A. Dyck, R. Penners, H. Lichter (2015): *Towards Definitions for Release Engineering and DevOps*. In 3rd International Workshop on Release Engineering (RELENG) associated with the 37th International Conference on Software Engineering (ICSE 2015), 19 May 2015, Florence, Italy, 3-3.

M. Vianden, H. Lichter, A. Steffens (2014): *Experience on a Microservice-based Reference Architecture for Measurement Systems*. In 21th Asia-Pacific Software Engineering Conference (APSEC 2014), Jeju, Korea, December 1-4, 2014, Vol. 1, 183-190.

M. Vianden, H. Lichter (2014): *Lessons Learned on Systematic Metric System Development at a large IT Service Provider*. In 2<sup>nd</sup> International Workshop on Quantitative Approaches to Software Quality (QuASoC), in conjunction with 21th Asia-Pacific Software Engineering Conference (APSEC 2014), Jeju, South Korea, December 1, 2014, Vol. 2, 47-40.

A. Dragomir, H. Lichter (2014): *Towards an Architecture Quality Index for the Behavior of Software Systems*. In 2nd International Workshop on Quantitative Approaches to Software Quality (QuASoC), in conjunction with 21th Asia-Pacific Software Engineering Conference (APSEC 2014), Jeju, South Korea, December 1, 2014, Vol. 2, 75-82.

A. Dragomir, H. Lichter, J. Dohmen, H. Chen (2014): *Run-time Monitoring-based Evaluation and Communication Integrity Validation of Software Architectures*. In 21th Asia-Pacific Software Engineering Conference (APSEC 2014), Jeju, South Korea, December 1-4, 2014, Vol. 1, 191-198.