SWC Software Construction

RWTH AACHEN UNIVERSITY

ANNUAL REPORT 2016

Contact
office@swc.rwth-aachen.de
www.swc.rwth-aachen.de
+49-241-80-21331
Ahornstr. 55
52074 Aachen, Germany

Software Construction

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

  Muhammad Firdaus Harun, M.Sc.
  (third-party funds position)

  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
  Dipl.-Inform. Matthias Vianden
  Tanya Sattaya-aphitan, M. Eng.

- **Student Researchers**
  Artjom Göringer
  Nora Nickel
  Yalcin Bogazci
  Anais Morel

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*
- *Cost Benefit based Technical Debt Management*
- *Model-based Software Architecture Evolution and Evaluation*
- *Regression Test Optimization*
- *DevOps-aware Software Engineering*
- *Architecture Roundtrip Management*
- *Domain Model-based Test Data Generation for Application Systems*

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)*
- *ARAMIS (Architecture Analysis and Monitoring Infrastructure)*

Last year, we organized the 3rd International Workshop on *Quantitative Approaches to Software Quality (QuASoQ)* which was held in conjunction with APSEC 2015 in New Delhi, India. Together with our partners from TM Munich and FAU Nurnberg we organized a workshop on *Continuous Software Engineering (CSE)*, held in conjunction with SE 2016 in Vienna, Austria.

Currently we are organizing the QuASoQ follow-up workshop co-located with APSEC 2016 in Hamilton, New Zealand. Additionally, we are busy in organizing the research track of XP 2017, to be held in Cologne in May 2017.

In December 2015, Tanya Sattaya-aphitan passed his Ph.D. defense on *An Automated White-Box Test Case Generation Framework for Web Applications* at the Thai German Graduate School, Bangkok Thailand.

Furthermore, in July 2016, Matthias Vianden successfully passed his Ph.D. defense on *Systematic Metric Systems Engineering: Reference Architecture and Process Model*. Congratulations to both!

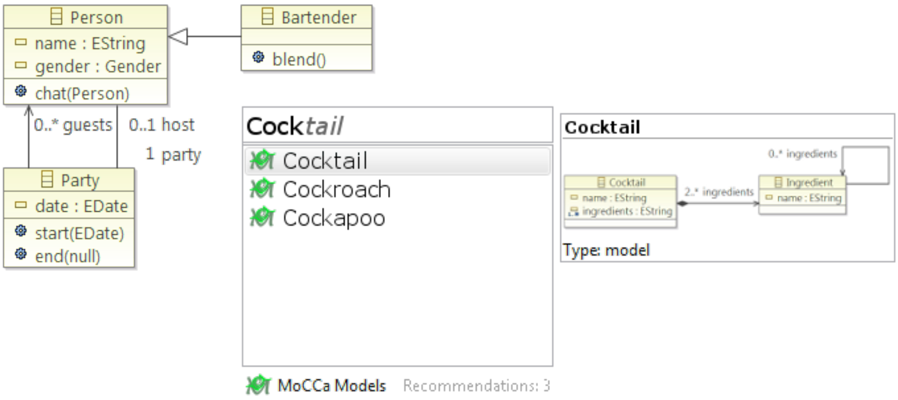
Since January 2016 we are one of the academic partners of the DAAD funded University-Business partnership project *Digital Lifestyle Germany-Malaysia*, where companies and universities in Germany and Malaysia have founded an international network, aiming to intensify the cooperation and to improve the education of the students of Universiti Teknologi Malaysia (UTM). In this context we gave a first block lecture at UTM in May 2016.

Research Projects

Reusing Domain Engineered Artifacts for Code Generation

A. Ganser, H. Lichter

Model driven architecture (MDA), and model driven engineering (MDE) take domain models as inputs for code generation, but only MDE includes reuse in domain specific modeling. 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.

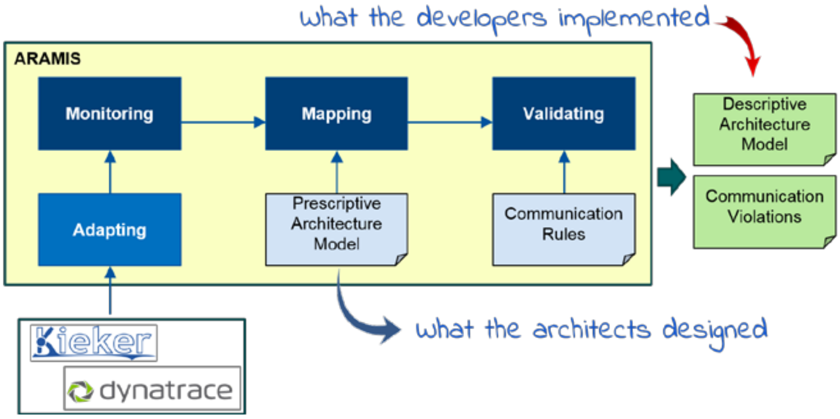
For more information please visit: http://hermes.modelrecommenders.org

Model-based Software Architecture Evolution and Evaluation

A. Nicolaescu, H. Lichter (Generali Informatik Services)

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 and evolution of software systems. With the advent of new architectural styles (component-based architectures, micro-services, etc.) there is a clear shift from mere structural towards behavior-based complexity. Consequently, ARAMIS focuses on the analysis and monitoring of the run-time of complex systems, 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.


what the developers implemented
ARAMIS
Monitoring
Mapping
Validating
Adapting
Prescriptive Architecture Model
Communication Rules
Descriptive Architecture Model
Communication Violations
Kieker
dynatrace
what the architects designed

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 analysis results can then be holistically visualized and/or further explored, by creating and applying behavior-centered views, viewpoints and perspectives. Furthermore, the architect can simulate changes to the prescriptive architecture description, observe their impact and compare various evolution strategies to make trade-offs explicit and support the choice of the most suitable ones.

Cost Benefit based Technical Debt Management

M. F. Harun, H. Lichter

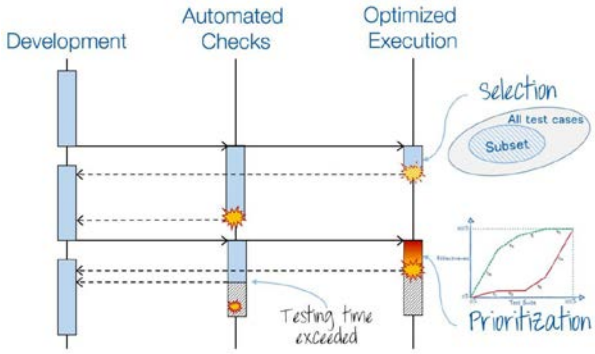
Structural weaknesses of code as well as of architectures can be manifested as a list of code smells/issues. This list can be ranked applying a code quality index (often called quality score). A low score indicates that the smell violates good design principle and may have a high negative impact to the system (e.g., in terms of maintainability). However, not all code smells having a low score value need to be refactored. Probably, the smell is less important from a business perspective or there is no harm to the architecture of the system for a long term. In addition, existing code quality management tools (such as Sonarqube, Codeclimate, Kiuwan etc.) provide a comprehensive review of the current state of a system but there is no support to take the decision when to refactor an existing smell (e.g., now or in a later release). Hence, project managers have to take these decisions only based on the score value of the smells.

Therefore, our approach, Cost-Benefit based Technical Debt Management (CoBeTDM), tries to support the refactoring decision of code smells using a dedicated cost-benefit model which aims to reduce the technical debt. The approach identifies the most critical smells based on current and historical change data. Then, it estimates the cost and benefit of possible refactorings for each smell. Finally, it determines the cost-benefit ratio to optimize the return of investment (spent in refactorings). This approach aims to help the project managers not only to improve the structural quality of a system but also to reduce the technical debt continuously.

Regression Test Optimization

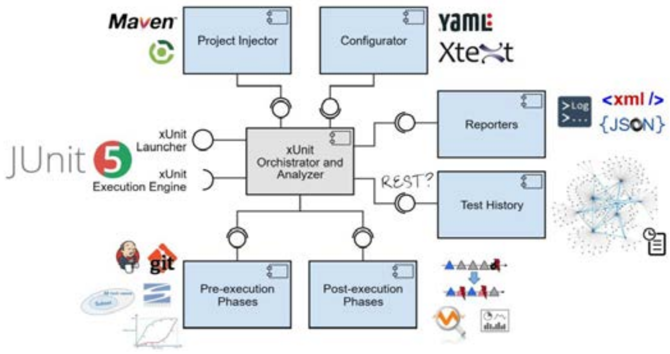
A. Dyck, H. Lichter

Software testing is by all means an important but also expensive software engineering tasks. Not only are automated checks a valuable tool to support software testers, but they also are essential for fast-evolving systems as well as continuous delivery done right.



With the growth of automated checks in a project, their execution time increases; thus, making fast feedback unfeasible. However, a change has only a partial impact on the system, one can choose to execute only relevant test cases (test selection). Further, the order of the test cases can reveal errors faster (test prioritization).

In research, several optimization techniques are proposed. However, those strategies often exists only on paper, don't scale, and tools are very rare to non-existent. This is partly due to the lack of extensibility of current check execution frameworks. Thus, we are working on an improved xUnit Check Execution Framework that allows for high modifiability and configurability; e.g., using any xUnit testing framework instance like JUnit or TestNG, integrating the framework via adapters in any project, configuring and extending pre- and post-check-execution phases, and providing an API for reports including a comprehensive test execution history.



DevOps-aware Software Engineering

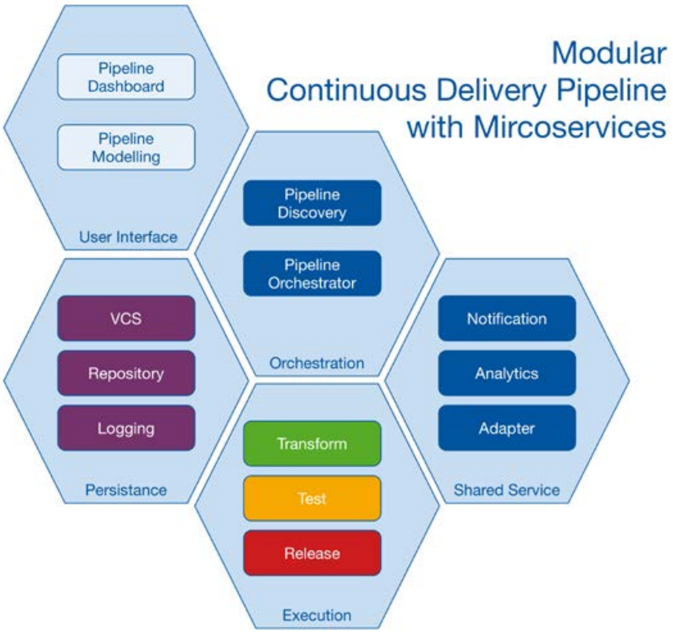
A. Steffens, H. Lichter (KISTERS AG and IVU AG)

The agile manifesto calls for the continuous delivery of valuable software to the customer. Based on this requirement a set of new approaches to software delivery were proposed, which lead to the concepts and methods of Continuous Integration and Delivery.

Both focus on the automation of building, testing, releasing and deploying of software. In addition these automation requires massive changes in the organization and culture of a software company. The term DevOps defines a mindset which includes these necessary changes on a technical, cultural and procedural level. A tight integration of methods from software engineering and software operation in the software development process can achieve the goal of building a continuous software delivery process.

This project aims to identify and evaluate a systematic approach of introducing methods from DevOps or CSE (Continuous Software Engineering) into the software development processes of two mid-sized software companies. Starting with the establishment of an organization wide deployment pipeline for all produced software components and infrastructure elements. At the end of this pipeline an executable system is delivered in terms of software, infrastructure, configuration and data.

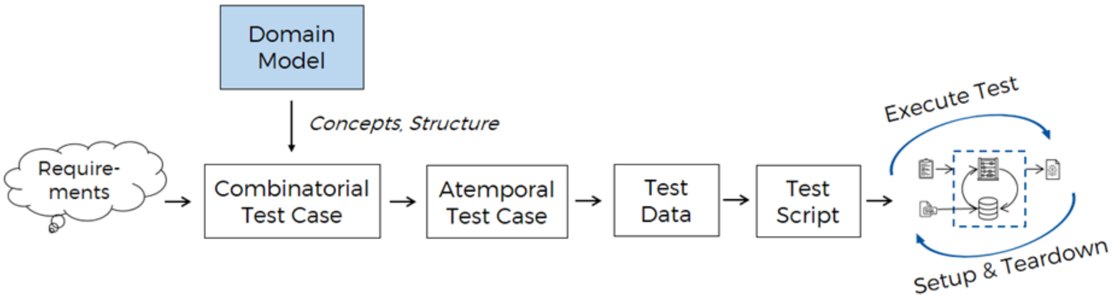
We focus on two very important components of a continuous software delivery process: the deployment pipeline itself and the approach to specify infrastructure elements as Infrastructure as Code (IaC). As the deployment pipeline is a complex software system we propose to apply best-practices from software engineering and created a modular microservice-based deployment pipeline. Handling infrastructure like computation resources in a declarative way as IaC enables teams to apply proven software engineering practices to the operation domain. In this project we investigate which practices can be applied and how to integrate IaC into the software development process.



Domain Model-based Test Data Generation for Application Systems

K. Fögen, H. Lichter (Generali Informatik Services)

Application systems such as enterprise resource planning systems or workflow management systems play important roles in today's companies as they increase productivity, reduce costs and support organizational change. Based on their importance and complexity, thorough testing is important. However, testing application systems is complex not only because of the number of different use cases but also because the same use cases are repeatedly tested with varying test data. In order to execute the tests with varying test data, it is necessary to design and generate the data prior to the test execution. This is a time-consuming and error-prone activity because much knowledge is needed to generate data which satisfies constraints enforced on the application level as well as temporal constraints enforced by the time of test execution.

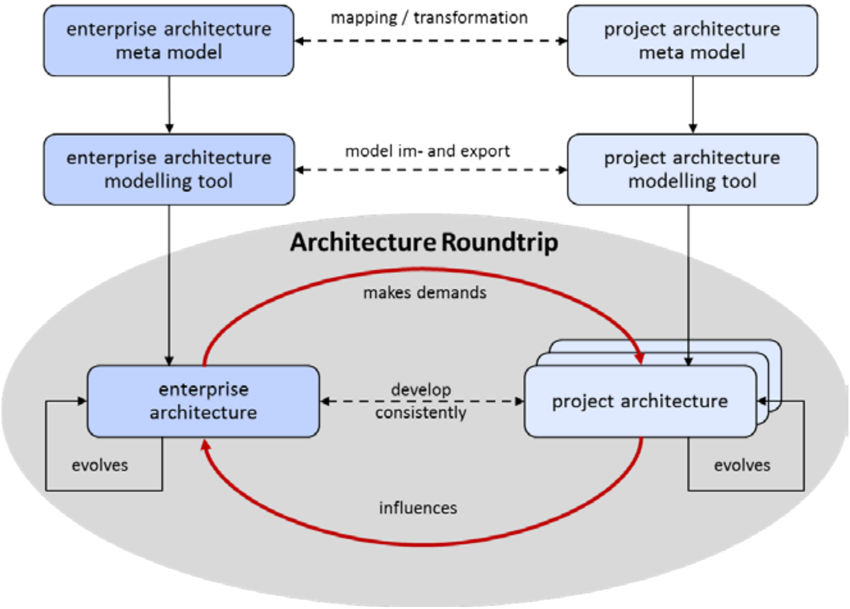


This research project aims to improve the activities of generating test data by systemizing related activities and by enhancing the reusability of previously generated artefacts. Therefore, we are researching approaches based upon the principles of domain-specific modeling and model-based testing to support test data generation in complex industrial settings.

Architecture Roundtrip-Management

S. Hacks, H. Lichter (ITERGO GmbH)

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, the project architectures often differ from the planned enterprise architecture. These changes have to transferred from the project architectures to the as-is enterprise architecture orchestrated by a quality management process. Additional it should be assessed how this 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 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 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

3rd International Workshop on Quantitative Approaches to Software Quality (QuASoQ)

New Delhi, India, December 6, 2015

QuASoQ 2015
3rd International Workshop on
Quantitative Approaches to Software Quality
New Delhi, India, Dec-1
in conjunction with
22nd Asia-Pacific
Software Engineering Conference
APSEC 2015

Collocated with the 22nd Asia-Pacific Software Engineering Conference (APSEC 2015), we organized the 3rd Int. Workshop "Quantitative Approaches to Software Quality" (QuASoQ).

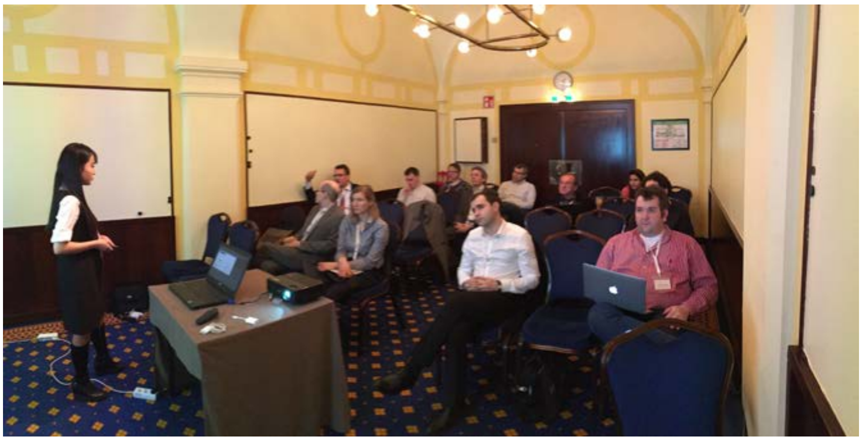
The goals of the QuASoQ workshop series are to exchange experiences, present new and promising approaches and discuss how to set up, organize, and maintain quantitative approaches to software quality.

More than 25 researchers attended the workshop where 6 papers are presented and discussed. The presented papers are published in a joint APSEC 2015 workshop proceedings at CEUR-WS.org.

1st Workshop on Continuous Software Engineering (CSE)

Vienna, Austria, February 23, 2016

Collocated with the Software Engineering conference (SE 2016), we organized a first workshop on "Continuous Software Engineering".



In order to develop and deliver high-quality products to their customers, software companies have to adopt state-of-the-art software development processes. To face this challenge, companies are applying innovative methods, approaches and techniques like agile methods, DevOps, continuous delivery, test automation, infrastructure as code or container-based virtualization.

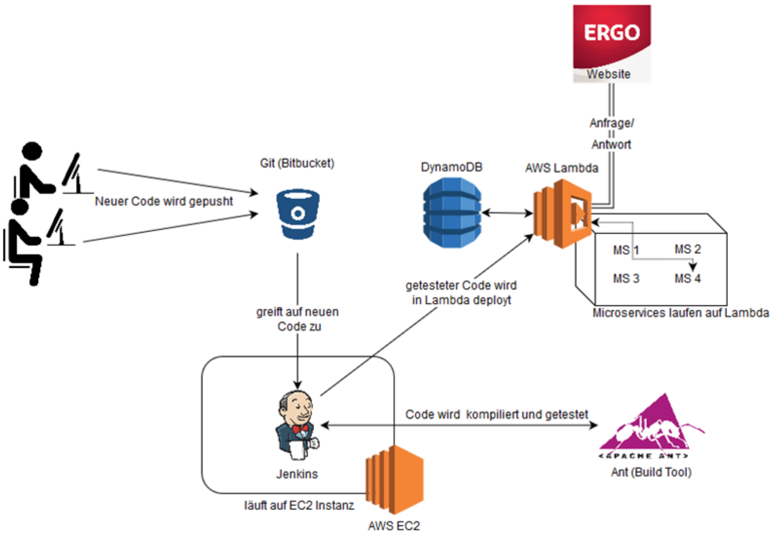
These new approaches have a high impact on the specification, design, development, maintenance, operation and the evolution of software systems. Therefore, common software engineering activities, organizational forms and processes have to be questioned, adapted and extended to ensure continuous and unobstructed software development (Continuous Software Engineering).

In this workshop 7 papers covering very different aspects of CSE have been presented and discussed. The presented papers are published in a joint SE 2016 workshop proceedings at CEUR-WS.org.

Tackling a real-world problem in a teaching lab

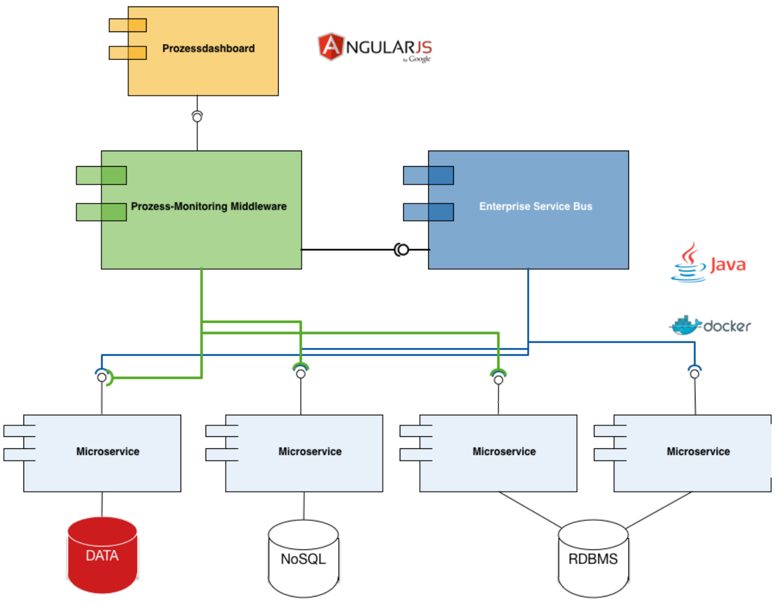
Due to our good experience in recent years, we decided to involve two of our industrial cooperation partners, ITERGO and KISTERS, in our software project lab once more.

ITERGO is the main IT supplier of the German ERGO insurance group. It offers everything from development, over IT-management to operations. Hence, ITERGO is historically grown and aligned to corporate group processes, its architecture sticks to mainframe and long running processes. To promote ITERGO to the new digital era, the students should develop a proof of concept, in which they show how to automate infrastructure generation in the Amazon cloud. Therefore, they developed some micro-services, stored them into Docker container and deployed them to the infrastructure.



KISTERS offers a broad portfolio of software solutions for companies in the energy sector. They proposed a complex business process to be implemented. The problem to tackle as suggested by KISTERS was to develop micro-services, which handle incoming energy market messages. To route the messages between the services an enterprise service bus was used, which was deployed as well as the services on an Openshift infrastructure. In addition, they implemented a dashboard, which gave a response to the users which service processed which message.

During the lab, our partners provided each a project manager who participated in discussions and approved certain milestones, as well as provided access to real data.



At the end of the project, each student team presented a working prototype and important insights into the problem domain in two small events hosted by our partners.

Many architects, developers and managers were attending these presentations and thrilled with the results the students were able to produce in one lab.

Other Activities

- Member of the international program committee, 31st ACM/SIGAPP Symposium on Applied Computing (SAC), Software Engineering Track, Pisa, Italy, April 4-8, 2016, *H. Lichter*
- Member of the international program committee, 2nd International Conference on Advances and Trends in Software Engineering (SOFTENG), February 21 – 25, 2016 Lisbon, Portugal, *H. Lichter*
- Member of the international program committee, 8th Asian Conference on Intelligent Information and Database Systems (ACIIDS 2016),14-16 March 2016, Da Nang, Vietnam, *H. Lichter*
- Organization and chair of the program committee, 1st International Workshop on Continuous Software Engineering (CSE), February 23, 2016, Vienna, Austria, *H. Lichter, A. Steffens*
- Organization and chair of the international program committee, 3rd International Workshop on Quantitative Approaches to Software Quality (QuASoQ), New Delhi, India, December 1, 2015, *H. Lichter, K. Fögen*
- Member of the international program committee, ICT International Student Project Conference, May 27-28, 2016, Bangkok, Thailand, *H. Lichter*
- Member of the organization team of “Tag der Informatik 2015”, *H. Lichter, A. Steffens*
- Reviewer for dpunkt-Verlag Heidelberg and computing reviews, *H. Lichter*
- Reviewer for Enterprise Modelling and Information Systems Architectures - An International Journal, *S. Hacks*
- 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 Advisory Committee of the International Networking in Science & Technology (INSTec), *H. Lichter*
- Lecturer for the “Kara, der programmierbare Marienkäfer” course at Helle Köpfe in der Informatik 2016, K. Fögen, *S. Hacks*
- Lecturer at Thai German Graduate School of Engineering (TGGS), course „Software Engineering“, Bangkok, Thailand, *H. Lichter*
- Lecturer in the context of the DAAD funded University-Business partnership project Digital Lifestyle Germany-Malaysia at UTM Johor Bahru, *H. Lichter, A. Steffens*
- Organization of the universal and specialized Preparatory Courses in Computer Science 2016, *H. Lichter, A. Dyck*

Talks and Publications

Talks

M. Vianden: *Systematic Metric Systems Engineering: Reference Architecture and Process Model.* Doctoral Seminar, July 20, 2016

M. Vianden: *Introduction to 3rd QuASoQ workshop*, Jeju, Korea, December 6, 2015.

Tanya Sattaya-aphitan: An Automated White-Box Test Case Generation Framework for Web Applications. Ph.D. defense talk, TGGS, Bangkok, Thailand, December 21, 2015

H. Lichter: *Software Architecture Evaluation based on Run-Time Monitoring*. Mahidol University, Salaya Campus, Bangkok, Thailand, November 20, 2015.

H. Lichter: *Behavior-based Architecture Reconstruction and Conformance Checking - The Meta-Model Incompatibility Problem*. 13th Working IEEE/IFIP Conference on Software Architecture (WICSA) 2016, Venice, Italy, April 6, 2016.

A. Steffens: *Continuous Software Engineering*. Introduction to the workshop CSE 2016, Vienna, Austria, February 22, 2016

A. Steffens: *Building Modular Deployment Pipelines with Microservices and Docker*. OODACH Workshop, Fraunhofer IAIS, St. Augustin, September 26, 2016.

A. Nicolaescu: *Checking Architectural Conformance based on Run-time Information*. OODACH Workshop, Fraunhofer IAIS, St. Augustin, September 26, 2016.

K. Fögen: Domain Model-based Test Data Generation for Testing Application Systems. Doctoral Symposium of the 2016 International Symposium on Software Testing and Analysis (ISSTA 2016), Saarland University, Saarbrücken, Germany, July 17, 2016

F. Harun: Towards a Technical Debt Management Framework based on Cost-Benefit Analysis. 10th International Conference on Software Engineering Advances (ICSEA 2015), Barcelona, Spain, November 16, 2015.

A. Ganser: Operation-Based Model Recommenders for MDE, University Hamburg, May 4, 2016.

Publications

A. Dragomir, H. Lichter (2016): *Behavior-based Architecture Reconstruction and Conformance Checking*. In 13th Working IEEE/IFIP Conference on Software Architecture (WICSA) 2016, Venice, Italy, April 5-8, IEEE, New York, NY, USA, 152-157.

V. von Hof, K. Fögen, H. Kuchen (2016): *Compilezeit-Prüfung von Spring-Konfigurationen*. In Gemeinsamer Tagungsband der Workshops der Tagung Software Engineering 2016 (SE 2016), Wien, CEUR-WS.org, Vol. 1559, 96-108.

W. Zimmermann, L. Alperowitz, B. Brügge, J. Fahsel, A. Herrmann, A. Hoffmann, A. Krall, D. Landes, H. Lichter, D. Riehle, I. Schaefer, C. Scheuermann, A. Schlaefer, S. Schupp, A. Seitz, A. Steffens, A. Stollenwerk, R. Weißbach (eds) (2016): *Gemeinsamer Tagungsband der Workshops der Tagung Software Engineering 2016 (SE 2016)*, Wien, 23.-26. Februar, 2016. CEUR-WS.org.

H. Lichter, B. Brügge, D. Riehle (2016): *Workshop on Continuous Software Engineering*. In Gemeinsamer Tagungsband der Workshops der Tagung Software Engineering 2016 (SE 2016), Wien, CEUR-WS.org, Vol. 1559, 136-137.

T. Sattaya-aphitan, H. Lichter, T. Anwar, S. Tanachutiwat (2016): *A meta-model for automatic modelling of dynamic web applications*. In Journal of Theoretical and Applied Information Technology, Little Lion Scientific, Vol. 84 (2), 203-214.

A. Ganser, H. Lichter, A. Roth, B. Rumpe (2016): *Staged model evolution and proactive quality guidance for model libraries*. In Software Quality Journal, Springer US, Vol. 24 (3), 675-708.

H. Lichter, T. Anwar, T. Sunetnanta, M. Vianden, A. Dubey, E. L. Celis, E. S. Grant, V. Shankararaman (eds) (2015): Joint Proceedings of the 3rd International Workshop on Quantitative Approaches to Software Quality (QuASoQ), the Workshop on Alternate Workforces for Software Engineering (WAWSE) and the 1st International Workshop on Case Method for Computing Education (CMCE) co-located with APSEC 2015, New Delhi, India, December 1, 2015. CEUR-WS.org.

F. Harun, H. Lichter (2015): *Towards a Technical Debt Management Framework based on Cost-Benefit Analysis*. In 10th International Conference on Software Engineering Advances (ICSEA 2015) November 15-20, Barcelona, Spain, 70-73.

L. Dung, A. Nicolaescu, H. Lichter (2015): *Adapting heterogeneous ADLs for software architecture reconstruction tools*. In 10th International Conference on Software Engineering Advances (ICSEA 2015) November 15-20, Barcelona, Spain, 52-55.

P. Alexander, A. Nicolaescu, H. Lichter (2015): *Model-Based Evaluation and Simulation of Software Architecture Evolution*. In 10th International Conference on Software Engineering Advances (ICSEA 2015) November 15-20, Barcelona, Spain, 153-156.