

# Report of the 2<sup>nd</sup> Workshop on Continuous Software Engineering

Stephan Krusche <sup>1</sup>, Horst Lichter <sup>2</sup>, Dirk Riehle <sup>3</sup> and Andreas Steffens <sup>4</sup>

## 1 Introduction

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.

After the successful 1<sup>st</sup> Workshop on Continuous Software Engineering (CSE 2016) [SEW16], held in Vienna, the organizers of the 2<sup>nd</sup> workshop wanted to widen the scope of the workshop. Therefore, the call for papers and the list of workshop topics were adjusted to the following:

### Processes and workflows

- Change management and handling user feedback
- Software development lifecycle for CSE
- Continuous delivery for requirements engineering
- Lean agile processes and practices

### Technologies and tools

- Infrastructure as code
- Provisioning of software and infrastructure
- Application virtualization with container
- Engineering of deployment pipelines

<sup>1</sup> Technische Universität München, Chair for Applied Software Engineering, krusche@in.tum.de

<sup>2</sup> RWTH Aachen University, Research Group Software Construction, lichtner@swc.rwth-aachen.de

<sup>3</sup> Friedrich-Alexander-University Erlangen-Nürnberg, Open Source Research Group, dirk.riehle@fau.de

<sup>4</sup> RWTH Aachen University, Research Group Software Construction, steffens@swc.rwth-aachen.de

*Copyright © 2017 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors.*

## 2nd Workshop on Continuous Software Engineering

### **Architecture**

- Design for scalability
- Software architecture for CSE
- Microservices
- Model driven architecture for CSE

### **Quality and testing**

- Test automation and optimization
- Monitoring and performance
- Security and Metrics for DevOps

### **Culture and business**

- Teaching CSE approaches
- Organizational issues for CSE
- Digital transformation and innovation

Overall, the workshop aimed at gathering together researchers and practitioners to present new ideas and discuss experiences in the application of state of the art approaches to Continuous Software Engineering.

## **2 Workshop Format**

Based on our former experience, we wanted the workshop to be highly interactive. In order to have an interesting and interactive event sharing lots of experience, we organized the workshop presentations applying the author-discussant model.

Based on this model, papers are presented by one of the authors. After the presentation, a discussant starts the discussion based on his or her pre-formulated questions. Therefore, the discussant had to prepare a set of questions and had to know the details of the presented paper. The general structure of each talk was as follows:

1. The author of a paper presented the paper (20 minutes).
2. After that, the discussant of the paper opened the discussion using his or her questions (5 minutes).
3. Finally, we moderated the discussion among the whole audience (5 minutes).

This format was very successful because it led to more intensive discussions among the participants.

### 3 Workshop Contributions

Altogether, eleven papers were submitted. Finally, six papers were accepted by the program committee for presentation and publication covering different topics. We grouped the papers into three sessions, added a special discussion session called “Industry meets Academia” and a final round-up session to discuss the major findings of our workshop. The following papers were presented:

#### Session A: Processes

- A1 Jan Ole Johanssen, Anja Kleebaum, Bernd Brügge and Barbara Paech: *Towards a Systematic Approach to Integrate Usage and Decision Knowledge in Continuous Software Engineering*
- A2 Martin Kleehaus, Ömer Uludag and Florian Matthes: *Towards a Multi-Layer IT Infrastructure Monitoring Approach based on Enterprise Architecture Information*

#### Session B: Industrial Experience

- B1 Masud Fazal-Baqaie, Baris Güldali and Simon Oberthür: *Towards DevOps in Multi-provider Projects*
- B2 Thomas Kurpick and Sebastian Melchior: *Naming in deployment pipelines for SaaS*

#### Session C: Techniques & Tools

- C1 Konrad Schneid: *Versioning strategies for developing new features within the context of Continuous Delivery*
- C2 Lukas Alperowitz, Andrea Marie Weintraud, Stefan Christoph Kofler and Bernd Brügge: *Continuous Prototyping: Unified Application Delivery from Early Design to Code*

### 4 Summary of the Discussions

In the following we summarize the main results of the discussions of the workshop.

#### Session A: Processes

The central idea of the approach presented in paper A1 is to capture decision knowledge for certain features by gathering feedback and usage data from continuous deployment. This data is combined with the initial feature request and the final decision to integrate the feature in order to create a knowledge repository, which can be then analyzed and queried in the further development. It was discussed that it is not obvious which information on which level of granularity is of interest for the stakeholders. Hence, the knowledge repository needs to be designed as adaptable as possible, especially if the

## 2nd Workshop on Continuous Software Engineering

existence of itself may lead to a changed behavior of developers and users which may lead to new information needs. In addition, the question arises if and how knowledge items evolve over time. The concept could be extended by introducing links or dependencies between knowledge items.

Paper A2 motivated that traceability between the elements of the different layers (business, application and infrastructure) of an Enterprise Architecture (EA) is a prerequisite to better understand the relationships as well as the impacts between these layers in case something changes. Moreover, there is a need to continuously evolve the business of an enterprise. It was discussed that dynamic trace information could be used to identify links between the elements of the EA layers. The high heterogeneity of technologies and data structures may prove a high hurdle to link EA items in different layers and it could be hard to match the identified links to the EA model elements in case there are deviations between the modelled EA and the implemented EA.

### **Session B: Industrial Experience**

Paper B1 reported on the results obtained to define a naming schema for components that support readability and understandability on the one hand. On the other hand, it has to be designed in a way that it does not violate naming constraints required by typical CI/CD tools. The discussion focused on the importance of a naming guideline and what could be automated in order not to violate the guideline and to apply the created names consistently in the process. Furthermore, it was discussed if there are more aspects that have mayor cross-cutting impacts on a CI/CD tooling infrastructure.

The presenter of paper B2 discussed interesting insights when running projects with several sub-contractors implementing features in parallel. The focus was on how to leverage these sub-contractors in a way that their performance becomes predictable and measurable. The subsequent discussion focused on the presented idea of applying dedicated maturity levels. Furthermore, it was discussed how to transfer knowledge to the teams involved using very different channels like videos, tutorials or documents.

In the closing “*Industry meets Academia*” session, the participants discussed the main activities of academia, teaching and research, especially how these activities should be adapted to fulfill the needs of the industry in the context of CSE. From industry, it was clearly stated that the whole application life cycle should be covered by education including all so called “continuous” activities. This could be achieved best, if lectures from universities and experts from the industry closely work together and if industry experts teach best practices, e.g. in guest lectures or if they participate in project courses. While there are already courses teaching CSE in university, they typically include smaller projects, as it is particularly challenging to create large project environments in the university. In addition, universities have to focus on the conceptual knowledge by teaching CSE abstractions and workflows, instead of focusing too much on concrete tools. From a research perspective, industry participants expected more usable and applicable knowledge, for example (anti-)patterns, best practices and processes to introduce CSE methods and techniques at an industry scale. As the bandwidth of tools is

## 2nd Workshop on Continuous Software Engineering

huge, the development of a conceptual framework for CSE activities was proposed, which could be used to instantiate CSE in a company while avoiding common errors and pitfalls.

### **Session C: Techniques and Tools**

Developing software in teams and in parallel can only be managed if applying an appropriate version control strategy. Paper C1 argued that feature development in the context of CI/CD could be organized applying feature toggles in the main branch or by using feature branches to separate the feature development into own branches; a hybrid approach could be feasible as well. The discussion shows that there is no silver bullet solution and the decision how to organize feature development in a CI/CD environment has to be taken carefully.

The last paper, paper C3, argued that non-executable prototypes, typically created by designers for user interfaces, have to be included in the CI/CD process, so that they are under control and can be used for systematic and frequent feedback from the users. Although prototyping is an excellent means to get early feedback for open questions, it has to be decided, when to stop prototyping and when to start the target development of the prototyped aspects. It was discussed, if this kind of prototyping could lead to expectations that cannot be fulfilled by the final implementation.

In the closing summary session, some topics were discussed that were not covered and where only little or no work has been published. One mayor topic was on key performance indicators for CSE, e.g. what metrics could be applied to assess the quality of CI/CD processes or artifacts like deployment pipelines. Those metrics could be used to monitor the evolution and to get early feedback if the quality of processes or artifacts decrease.

Another discussed topic was, how universal CI/CD approaches are regarding the different software development domains and their specific requirements. The question appeared, what is common for the different CI/CD approaches applied in industry. In this context, solutions for integrating existing legacy systems into CI/CD processes are needed.

We want to summarize the central results of the workshop by proposing some key challenges for CSE research and education in the future.

- Develop a conceptual framework for CSE methods, techniques and processes
- Derive best practices and patterns from real life experience
- Develop metrics and KPIs to assess CSE and its implementation.
- Extend SE teaching by including CSE topics and by cooperating closely with industry

## 2nd Workshop on Continuous Software Engineering

Based on the results of the workshop the organizers agreed to propose a follow-up workshop in 2018. In this follow-up workshop, the organizers want to facilitate more industry contributions by lowering the barrier for industry presentations. Furthermore, some ideas were discussed to initiate and establish a working group on CSE in the context of the Germany society of informatics (GI). The organizers and also some industry participants confirmed to join this working group as founding members.

## 5 Acknowledgements

Many people contributed to the success of this workshop. First of all, we want to give thanks to the authors and presenters of the accepted papers. Furthermore, we want to express our gratitude to the SE 2017 organizers for supporting our workshop. Finally, we are glad that these people served on the program committee, soliciting papers and writing peer reviews:

- Lukas Alperowitz TU München
- Jan Bosch Chalmers University of Technology
- Michael Goedicke University of Duisburg-Essen
- Willi Hasselbring Universität Kiel
- Martin Jung develop group, Erlangen
- Stephan Krusche TU München (Organizer)
- Horst Lichter RWTH Aachen University (Organizer)
- Christian Nester Google Inc.
- Dirk Riehle FAU Nürnberg (Organizer)
- Heinz-Josef Schlebusch Kisters AG, Aachen
- Andreas Steffens RWTH Aachen University (Organizer)
- Christian Uhl codecentric AG, Düsseldorf
- Andre von Horn Universität Stuttgart
- Stefan Wagner Universität Stuttgart
- Heinz Züllighoven WPS und Universität Hamburg

## References

- [SEW16] 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. Schlaefel, 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., Vol-1559.