

Joint Proceedings of the Summer School on Software Maintenance and Evolution

Editorial Board Members

Nyyti Saarimäki
Valentina Lenarduzzi
Davide Taibi

Preface

This volume contains the joint proceedings of two Inforte Summer Schools on software maintenance and evolution arranged during the summer 2019: “[Software Quality and Bug Prediction with Machine Learning](#)” organized on June 10-12 2019 and “[Summer School on Software Evolution: From Monolithic to Cloud-Native](#)” organized on September 2-4 2019. [Inforte](#) is a Finnish state wide programme for ICT professionals. It is designed to offer networking and education events to PhD students and professionals working in Finnish companies, polytechnics and public administration.

Program Committee

Fabio Palomba (University of Zurich)
Sebastiano Panichella (Zurich University of Applied Science)
Josef Spillner (Zurich University of Applied Science)
Valentina Lenarduzzi (Tampere University)
Davide Taibi (Tampere University)
Kari Systä (Tampere University)
Outi Sievi-Korte (Tampere University)
David Hästbacka (Tampere University)

Summer School on Software Quality and Bug Prediction with Machine Learning

June 10 –12, 2019, Tampere, Finland

The aim of the seminar was to familiarize the participants with different aspects of software quality and different machine learning methods that can be used to predict it. The participants analyzed the evolution and maintenance of software projects using machine learning and a given dataset. The dataset provided the version history of technical debt issues for more than 30 open source software projects at commit level, including information on commits, faults, technical issues and many more. The analyses could be based on this dataset alone or expanded to also include data from other resources such as GitHub. The overall goal was to study the evolution and maintenance of the projects in the dataset.

At the end of the seminar, the participants were asked to finalize a paper on their project work. The paper were peer-reviewed.

Invited Speaks

Software Design 101: Improving the Design of Existing Code, Tests, and Communities

Fabio Palomba, University of Zurich, Switzerland

In 1999, Martin Fowler introduced the concept of code smells, describing them as symptoms of poor implementation choices that may possibly worsen source code design and induce software faults. Besides keeping source code design under control, the continuous evolution of software systems and the technologies associated with them - from Continuous Integration to DevOps - has had the effect of putting in the spotlight tests and people, that are now more than ever required to work in synergy with production code for the successful evolution of software systems. In this talk, I will overview the state of the art with respect to empirical software engineering and mining software repository methods that allow (i) the investigation and control of source code, test code, and development community design and (ii) the analysis of the interactions among them.

Machine Learning Techniques for Bug Prediction

Valentina Lenarduzzi, Tampere University, Finland

Bug Prediction is one of software engineering's holy grails, which concerns with the overall of software successes. Predicting the software faults already from the earlier phase improves the software quality, reliability, efficiency and reduces the maintenance cost. Developing robust bug prediction model is a challenging task and many techniques have been proposed in the literature. In this session, we will overview the state of the art of bug prediction model focusing on machine learning techniques, looking at the selection, application and data management. In this session we will provide a set of options to help students to select the most appropriate methods for their post seminar challenge.

Summer School on Software Evolution: From Monolithic to Cloud-Native

September 2 –4, 2019, Tampere, Finland

Preface

The aim of the [seminar](#) was to familiarize the participants with software evolution in the cloud. Participants were invited to submit papers on cloud-native system, software evolution in the cloud, microservices and serverless. Papers were peer-reviewed by the program committee

Invited speaks

Cloud Native Migration Patterns

Davide Taibi

Cloud-Native technologies, and especially microservices are enjoying increasing popularity and diffusion in industrial environments, being adopted by several big players such as Amazon, LinkedIn, Netflix, and SoundCloud. Several patterns and platforms such as nginx (www.nginx.org) and Kubernetes (kubernetes.io) exist on the market. During the migration process, practitioners often face common problems, which are mainly due to their lack of knowledge regarding bad practices and patterns. In this session, we provide an introduction of the cloud-native patterns, reporting their issues and motivations, and describing the most common patterns and anti-patterns.

Cloud-based Testing

Sebastiano Panichella, University of Zurich, Switzerland

To promote and sustain the future of our society, the most critical challenge of contemporary software engineering and cloud computing experts are related to the efficient integration of emerging cloudification and DevOps practices in the development and testing processes of modern systems.

In this context, we argue that Search-based Software testing (SBST) can play a critical role in improving testing practices and automating the verification and validation (V&V) of cloudification properties of Cloud Native Applications (CNA). We, we focus on the untouched side of SBST in the cloud field, by discussing (1) the testing challenges in the cloud research field and (2) summarizing the recent contributions of SBST in supporting development practices of CNA. Then, we discuss the emerging research topics characterizing the cloudification perspectives of SBST in the cloud field. Finally, we provide evidence on how SBST can be used to improve testing practices in the context of cloud-native applications.

Software evolution in microservice artefacts

Josef Spillner, Zurich University of Applied Sciences, Switzerland

Modern cloud software applications are characterised by mixed-technology compositions of stateless and stateful artefacts, many of them representing microservices. Through open marketplaces, both generic and artefact type-specific ones, it is possible to learn about the evolution of artefacts and microservices over time. Specifically, metrics related to the supply and demand side - published and downloaded or deployed artefacts - and technical metrics related to code and configuration are already measured in long-term experiments by researchers. This talk outlines current artefact observations, identifies common trends and diverging metrics, and explains how researchers can exploit the information gained from the evolution to help developers build better cloud software.

Kubernetes, Operators SDK, and Prometheus

Sebastiano Panichella, University of Zurich, Switzerland

Emerging technologies are used to perform the verification and validation (V&V) of cloudification properties of Cloud Native Applications (CNA) as well as to enable a better observability, monitoring, and management of these applications. In this talk we provide some hands-on and practical insights on the most emerging technologies characterizing the emerging cloud computing and software engineering research: Kubernetes, Operators SDK, and Prometheus. Then, we discuss the emerging research topics characterizing the usage of such tools the context of cloud-native applications.

Cloud service dependencies in practice

Josef Spillner, Zurich University of Applied Sciences, Switzerland

In theory, software and service dependencies are simple. They are simple to define through declarative files, simple to resolve by package managers and service brokers, and simple to understand. In practice, they are hard. Dependencies are often implicit, nested or transitive, conflicting, and moving targets. This talk first conveys general knowledge about dependencies, presents algorithms for deterministic and heuristic dependency detection, and then explores them in popular composite (orchestrated) application formats. The talk also touches on the related topic of software application portability. 11:30 - 12:30 “Students presentation” students will make a short presentation (10 min per student) where they will present 1) their Ph.D. topics, 2) the research questions they are planning to investigate 3) the research techniques they are planning to apply