Message from the Invited Speakers Co-chairs
ASWEC 2015

It is a common requirement in academic software engineering conferences that all submissions are original and have not been published or presented previously. This has the unfortunate side-effect that even the best work in software engineering research gets presented only once, unless the presenters are invited subsequently to present the same work as part of a keynote or invited presentation.

Inspired by the Computer Science Education Conventions in Australia, it was decided to add an invited paper track to ASWEC 2015 in which prominent Australasian researchers who published major works in top international software engineering venues in the past 1-2 years to present that work again at ASWEC. As a result, the workload for the authors, presenters, and track chairs is relatively lightweight, but we hope and believe that there will be significant benefits to both the presenters and the conference audience. We hope you enjoy the presentations in this track and welcome your feedback.

Paul Strooper, *Queensland University, Australia*
Liming Zhu, *NICTA, Australia*

*ASWEC 2015 Invited Speakers Co-chairs*
Invited Speakers

Developing and Evaluating Software Engineering Process Theories

Paul Ralph
University of Auckland, New Zealand

Paper from Proceedings of
2015 IEEE/ACM 37th IEEE International Conference on Software Engineering,
Florence, Italy, IEEE, Volume 1, pp. 20-31

Abstract
A process theory is an explanation of how an entity changes and develops. While software engineering is fundamentally concerned with how software artifacts change and develop, little research explicitly builds and empirically evaluates software engineering process theories. This lack of theory obstructs scientific consensus by focusing the academic community on methods. Methods inevitably oversimplify and over-rationalize reality, obfuscating crucial phenomena including uncertainty, problem framing and illusory requirements. Better process theories are therefore needed to ground software engineering in empirical reality. However, poor understanding of process theory issues impedes research and publication. This paper therefore attempts to clarify the nature and types of process theories, explore their development and provide specific guidance for their empirically evaluation.

Speaker’s Bio
Dr. Paul Ralph is an author, scientist, consultant and computer science lecturer at The University of Auckland. His research centers on the theoretical and empirical study of software and game development, including projects, processes, practices, tools and developer cognition, socialization, productivity, creativity, wellbeing and effectiveness. His research has been published in premier software engineering and information systems outlets, including the International Conference on Software Engineering (ICSE), the International Conference on Information Systems (ICIS), the Journal of the Association for Information Systems (JAIS) and Information and Software Technology. Additionally, he has written editorials on technology, education and design for influential outlets including Business Insider, Lifehacker and The Conversation. Paul co-founded the AIS Special Interest Group for Game Design and Research (SIGGAME) and chaired the 4th International Workshop on General Theories of Software Engineering (GTSE). Paul holds a PhD in information systems from the University of British Columbia. Previously he was a lecturer at the Lancaster University Management School, the highest rated management research institution in the United Kingdom.
How Much Up-Front? A Grounded Theory of Agile Architecture

Michael Waterman, James Noble, George Allan
Specialised Architecture Services Ltd., New Zealand; Victoria University of Wellington, New Zealand

Paper from Proceedings of

Abstract
The tension between software architecture and agility is not well understood by agile practitioners or researchers. If an agile software team spends too little time designing architecture up-front then the team faces increased risk and higher chance of failure; if the team spends too much time the delivery of value to the customer is delayed, and responding to change can become extremely difficult. This paper presents a grounded theory of agile architecture that describes how agile software teams answer the question of how much upfront architecture design effort is enough. This theory, based on grounded theory research involving 44 participants, presents six forces that affect the team’s context and five strategies that teams use to help them determine how much effort they should put into up-front design.

Speaker’s Bio
James Noble is Professor of Computer Science and Software Engineering at Victoria University of Wellington, New Zealand. His research centers around software design. This includes the design of the users' interface, the parts of software that users have to deal with every day, and the programmers' interface, the internal structures and organisations of software that programmers see only when they are designing, building, or modifying software. His research in both of these areas is coloured by my longstanding interest in object-oriented approaches to design, and topics he has studied range from aliasing and object ownership, design patterns, agile methodology, via usability, visualisation and computer music, to postmodernism and the semiotics of programming.
Empirical Study Towards a Leading Indicator for Cost of Formal Software Verification

Daniel Matichuk, Toby Murray, June Andronick, Ross Jeffery, Gerwin Klein, Mark Staples
(Accepted)
NICTA, Australia; University of New South Wales, Australia

Paper from Proceedings of

Abstract
Formal verification can provide the highest degree of software assurance. Demand for it is growing, but there are still few projects that have successfully applied it to sizeable, real-world systems. This lack of experience makes it hard to predict the size, effort and duration of verification projects. In this paper, we aim to better understand possible leading indicators of proof size. We present an empirical analysis of proofs from the landmark formal verification of the seL4 microkernel and the two largest software verification proof developments in the Archive of Formal Proofs. Together, these comprise 15,018 individual lemmas and approximately 215,000 lines of proof script. We find a consistent quadratic relationship between the size of the formal statement of a property, and the final size of its formal proof in the interactive theorem prover Isabelle. Combined with our prior work, which has indicated that there is a strong linear relationship between proof effort and proof size, these results pave the way for effort estimation models to support the management of large-scale formal verification projects.

Speaker’s Bio
Dr Mark Staples is a Principal Researcher at NICTA, Australia’s ICT Research Centre of Excellence, and a Conjoint Associate Professor at the UNSW School of Computer Science and Engineering. He has held software and systems engineering leadership roles in teams within multi-national industrial companies in Australia, and at Australian startup companies. At NICTA he has led the Empirical Software Engineering research group, and also led the creation of the Future Logistics Living Lab jointly with Fraunhofer IESE and SAP Research. His current research is on software engineering with formal methods, to support assurances for trusted software-based systems. He has undergraduate degrees from The University of Queensland, and a PhD from the University of Cambridge Computer Laboratory.
Chiminey: Reliable Computing and Data Management Platform in the Cloud


Applied Data Science, Australia; RMIT University, Australia; Monash University, Australia

Paper from Proceedings of

Abstract

The enabling of scientific experiments that are embarrassingly parallel, long running and data-intensive into a cloud-based execution environment is a desirable, though complex undertaking for many researchers. The management of such virtual environments is cumbersome and not necessarily within the core skill set for scientists and engineers. We present here Chiminey, a software platform that enables researchers to (i) run applications on both traditional high performance computing and cloud-based computing infrastructures, (ii) handle failure during execution, (iii) curate and visualize execution outputs, (iv) share such data with collaborators or the public, and (v) search for publicly available data. Demo video: http://youtu.be/Twi-d2WT94A

Speaker’s Bio

Heinz Schmidt is Professor of Software Engineering in RMIT’s School of Computer Science and IT. He directs eResearch and the Australia-India Center of Automation Software Engineering (AICAUSE). Heinz received his PhD from Bremen University, Germany. Heinz is also Adjunct Professor at Mälardalen University, in Västerås (near Stockholm) in Sweden, where he works with the Mälardalen Real-Time Research Centre and the Embedded Systems Group in their CS department. Prof Schmidt is internationally recognised in software engineering and architecture for parallel and distributed systems. He has over 30 years experience with object-oriented and component-based software architectures, languages, systems and tools, in practice, research and training. Before joining RMIT, Heinz held positions at Monash University, the CSIRO, ANU, the International Computer Science Institute of UC Berkeley and the German Research Centre for Mathematics and CS (now transitioned into Fraunhofer). Over the last few years his research has particularly focussed on modeling, predicting and verifying extra-functional properties in software-intensive distributed platforms. Such properties include availability, reliability, performance and safety. Heinz has led industry collaborations in Australian CRC contexts and European projects.
StressCloud: A Tool for Analysing Performance and Energy Consumption of Cloud Applications

Feifei Chen, John Grundy, Jean-Guy Schneider, Yun Yang, Qiang He
Swinburne University of Technology, Australia

Paper from Proceedings of

Abstract

Finding the best deployment configuration that maximises energy efficiency while guaranteeing system performance of cloud applications is an extremely challenging task. It requires the evaluation of system performance and energy consumption under a wide variety of realistic workloads and deployment configurations. This paper demonstrates StressCloud, an automatic performance and energy consumption analysis tool for cloud applications in real-world cloud environments. StressCloud supports 1) the modelling of realistic cloud application workloads, 2) the automatic generation and running of load tests, and 3) the profiling of system performance and energy consumption. A demonstration video can be accessed at: https://www.youtube.com/watch?v=0l4_a_CNtVQ

Speaker’s Bio:

Jean-Guy Schneider has a M.Sc. and Ph.D. in Computer Science and Applied Mathematics from the University of Berne, Switzerland. He is an Associate Professor in Software Engineering at Swinburne University of Technology and currently the Associate Dean (Research and Engagement) of the Faculty of Science, Engineering and Technology (FSET) at Swinburne. Broadly, his research lies in the general area of reliable software systems and are positioned in the intersection of Software Engineering and Computer Science. More specifically, his research interests are in object-oriented and concurrent/distributed/service-oriented systems, programming languages, and the definition of formal approaches for component-based Software Engineering. Furthermore, he is interested in methodologies and tools in the context of the evolution of object- and component-based software systems, agile software development processes, mobile computing, as well as the influence and applicability of software development processes in tertiary education. He has extensively published in internationally leading journals and conferences. Jean-Guy has contributed to the Australasian Software Engineering research community in a variety of ways. Most notably, he was the General Chair of the 22th ASWEC in 2013, is a member of the ASWEC Steering Committee (in the role of past General Chair), and he is the Co-Chair of the 2015 ASWEC Doctoral Symposium.