

The Future of Software engineering IN and FOR the Cloud

Rami Bahsoon, The University of Birmingham, UK (r.bahsoon@cs.bham.ac.uk)

Ivan Mistrik, Independent Consultant, Germany

Nour Ali, The University of Brighton, UK

T.S. Mohan, Infosys Technologies, India

Nenad Medvidović, University of Southern California, USA

Experts are predicting the cloud software market to grow at a high rate in the coming years. Merrill Lynch and Gartner predict the cloud computing market to be worth between \$160 and \$150 billion in 2013. Many computing services have now migrated to the cloud and many applications are becoming cloud-based and function as standalone or in orchestration with other cloud-based services. With the advent of the cloud and the reliance on Big Data, cloud-based systems are driving our digital economy and increasingly becoming pervasive in many daily applications. Cloud has become the backbone for many e-businesses; engineering, scientific and experimental processes; real-time control and monitoring data intensive systems such as disaster management, weather and climate predictions, which heavily depends on the Big Data thing. The dependability of these systems and their continuous evolution are essential pre-requisite for the success and prosperous growth of our digital economy. The “cloud wave” has urged the need for advancing the fundamentals of cloud software engineering, supporting the emergence and the advancement of this emergent digital economy.

Practitioners are very likely to beg, borrow and steal from software engineering in-the-small to benefit the case of software engineering for the large-scale and ultra-large-scale, as it is the case of the cloud. Though the fundamentals of engineering software in both paradigms exhibit resemblance, software engineering for the cloud require novel approaches, which address the interplay between technical, economics-driven considerations and shifts software engineering towards utility-based engineering for software-, infrastructure-, data storage- and/or platform- as services. Given the unpredictable, dynamic, elasticity and on-demand nature of the cloud, it would be unrealistic to assume that traditional software engineering can “cleanly” satisfy the cloud behavioural requirements. Much of cloud software engineering, for examples, involves problems related to designing for emergent behaviour and engineering for scale and its economies in unpredictable, volatile and highly uncertain world. The paradigm has introduced new challenges and problems shaping the scientific and engineering queries and likely to shape the research landscape of software engineering for at least the next decade: cloud requirements engineering; architecting for the cloud; cloud security software engineering; service-level Agreement(SLA) compliance; maintenance, testing and evolution for the cloud; virtualisation; autonomic and self-* management for scale; cloud as a market place and economics-driven software engineering for the cloud; multi-tenancy and federation; dependability and performance engineering for the cloud; allocation assignment and dynamic balancing and optimisation for resources to achieve pragmatic trade-offs between many competing technical , short-term and strategic business objectives.

The goal of this special issue is to strengthen the cross-fertilization of advances from software engineering, services and cloud computing. The special issue aims at exploring, debating and increasing our understanding to the following:

- (i) how advances in software engineering, with emphasis on engineering requirements software architectures, architecting dependable systems, self-adaptive software architectures, economics-driven software engineering, utility computing, risk management, security software engineering and testing, Search-based software engineering can (not) benefit the case of cloud;
- (ii) what are the most recent innovations, trends, experiences and concerns in the field that appraise the paradigm-shift in engineering software systems as cloud services or in support of cloud infrastructures;
- (iii) What are the open research challenges and promising directions for software engineering FOR the cloud? And how cloud is likely to shape the research landscape of software engineering for at least the next decade?
- (iv) How the paradigm will shape the future of engineering software IN the cloud, i.e. benefiting from the cloud infrastructure, virtualization and economies of scale?

The special issue aims to bridge the gap between software engineering, services, business and cloud computing communities by specifically addressing the challenges for software engineering FOR and IN the cloud. The special issue extends our scientific query for probing an answer for the above through successive workshops on the Future of Software engineering IN and FOR the Cloud in conjunction with IEEE Cloud, IEE ICWS, and IEEE SCC in conjunction with IEEE Congress on Services, the flagship conference on cloud and service computing.

The special features some invited contributions from key researchers and practitioners in cloud software engineering for road mapping directions in the field. It compiles a number of academic and industrial columns reflecting on the state-of-arts and –practice of cloud software engineering. In addition to the invited contributions, our open call had solicited some novel (industrial and academic) regular manuscripts in the broad area of Cloud Software Engineering. Each was rigorously reviewed by at least three independent reviewers, in two phased-selection process. Invited columns had benefited from shepherding rigorous reviews. All articles were selected based on their originality, quality, and relevance to the theme.

A representative list of topics includes but not limited to:

- Requirements engineering in and for the cloud;
- Relating non-functional requirements to architectures for cloud environments;
- Architecting for the cloud;
- Patterns and architectural styles for the cloud;
- Agile software development on the cloud;
- Engineering security, trust and privacy in architectures for cloud environments;
- Engineering for performance, reliability, heterogeneity, safety, scalability, real-time and dependability in cloud architectures;
- Service-level management for the cloud;
- Mobile Services engineering for the cloud;
- Model-driven engineering for the cloud;
- Environments and tools support for the cloud;
- Testing for the cloud;
- Maintenance and evolution for the cloud;
- Risk management in the cloud;
- Engineering sustainability in cloud architectures;

- Economics-driven engineering for the cloud;
- Cloud as utilities;
- Search-based software engineering in/for the cloud;
- Empirical and industrial studies;
- Cloud software engineering education.

The special issue opens with an invited article by Mark Harman et al on Cloud engineering is Search Based Software Engineering too. Harman et al. argue that many of the problems posed by the migration of computation to cloud platforms can be formulated and solved using techniques associated with Search Based Software Engineering (SBSE). Much of cloud software engineering involves problems of optimisation: performance, allocation, assignment and the dynamic balancing of resources to achieve pragmatic trade-offs between many competing technical and business objectives. SBSE is concerned with the application of computational search and optimisation to solve precisely these kinds of software engineering challenges. Hraman's et al. article focuses on the ways in which SBSE can benefit cloud computing and introduce the theme of SBSE *for* the cloud, formulating cloud computing challenges in ways that can be addressed using SBSE.

A goal-oriented simulation approach for obtaining good private cloud-based system architectures Lawrence Chung et al., proposes a goal-oriented simulation approach for cloud-based system design whereby stakeholder goals are captured, together with such domain characteristics as workflows, and used in creating a simulation model as a proxy for the cloud-based system architecture. Simulations are then run, in an interleaving manner, against various configurations of the model as a way of rationally exploring, evaluating and selecting among incrementally better architectural alternatives. The authors then illustrate the approach using a smartcard-based public transportation system deployed in a private cloud. Using this approach, architects can more rationally and systematically transition from stakeholder goals to the architectural design of a cloud computing-based system.

The reflection article, **Cloud computing security: The scientific challenge, and a survey of solutions** by **Mark D. Ryan** surveys issues in cloud security engineering. The fact that data are shared with the cloud service provider is identified as the core scientific problem that separates cloud computing security from other topics in computing security. Ryan surveys approaches to protecting data from a cloud infrastructure provider. He reflects on three current research directions to solve the problem, and evaluate them using software-as-a-service example. Ryan explains some difficulties with using fully homomorphic encryption for cloud computing applications. He then describes a method in which in-browser key translation allows a software-as-a-service application to run with confidentiality from the service provider explore how trusted hardware can be used to protect cloud-based data.

An authentication model towards cloud federation in the enterprise by Nouredine and Basharoush, reports on the optimisation of identity federation in the cloud Marketplace. This optimization is achieved by introducing provisioning steps to pre-establish trust amongst enterprise applications' Resource Servers, its associated Authorization Server and the clients interested in access to protected resources. In this architecture, trust is provisioned and synchronized as a pre-requisite step to authentication amongst all communicating entities in OAuth protocol, and referral tokens are used to establish trust federation for Marketplace applications across organizations. A real-life case study and a simulation test were used to validate the results.

A framework to support selection of cloud providers based on security and privacy requirements by Mouratidis et al. provide a systematic and structured approach that enables software engineers to identify security and privacy requirements and select a suitable cloud service provider based on such requirements. The framework can help organisation to identify and analyse security, privacy constraints, security and privacy goals, threats and vulnerabilities relevant to a cloud based system.

A service-oriented framework for developing cross cloud migratable software, by Joaquín Guillén, Javier Miranda, Juan Manuel Murillo and Carlos Canal presents a framework for developing cloud agnostic applications that may be deployed indifferently across multiple cloud platforms. The framework tackles the development scale by separating and managing Information about cloud deployment and cloud integration from the source code. Interoperability between interdependent components deployed in different clouds is achieved by automatically generating services and service clients. This allows software developers to segment their applications into different modules that can easily be deployed and redistributed across heterogeneous cloud platforms.

A common API for delivering services over multi-vendor cloud resources by Luís Bastião, Carlos Costa, José Luís Oliveira observes that cloud players tend to create new services with different APIs, which imply that cloud-oriented applications might be instantiated in one single cloud provider. This scenario is not desirable to the IT industry because their applications will become provider-dependent. The authors then present a platform that allows applications to interoperate with distinct cloud providers' services using a normalized interface. The proposed approach provides a common API that minimizes the present deficit of cloud API standardization and provides secure and redundant services allocation.

The reflection column by **Tommi Mikkonen and Antero Taivalsaari on "Cloud computing and its impact on mobile software development: Two roads diverged"** argues that in the 2010's we will witness a major battle between two types of technologies: native web apps and Open Web applications that run in a web browser. This "Battle of the Decade" will determine the future of the software industry for years to come. The former approach implies the use of binary software and traditional software engineering practices, while the latter approach implies that conventional software engineering methods and practices will be replaced by web technologies. The position reflects on what the cloud can offer to mediate the battle.

The reflection position on **Solidifying the Foundations of the Cloud for the Next Generation Software Engineering by Vazquez-Poletti, Moreno-Vozmediano, Montero, Huedo and Llorente** argues that clouds are "too" infrastructure-oriented and they lack advanced service oriented capabilities such as service elasticity, quality of service or admission control to perform a holistic management of cloud-based systems and to deliver dependability. The authors argue that leveraging on service-oriented capabilities and strengthening the abstraction of various cloud layers can be a solution. The authors discuss a number of research directions to bridge this gap.

The industrial column on **Industry's role in data and software curation in the cloud by Judith Bishop, Microsoft Research**, explores how a researcher painlessly leverages the cloud to archive, share and publish scientific data. Bishop coins the phrases "middleware and baseware" as they apply to the cloud and give two examples of middleware that is freely available for data and for software curation and sharing. Bishop closes her reflection column with some ideas about the future of the cloud from the industry-researcher standpoint.

Last but not least, the special issue concludes with an invited article on **Teaching cloud computing: A software engineering perspective by Ian Sommerville**. Sommerville argues that the cloud is nothing new for software engineering. The fundamental issues of managing problem and solution complexity remain the same and it will always be very difficult for students to understand and appreciate these. Sommerville adds that the cloud means that we can at least partially address the problems of teaching about scale and we do have a responsibility to ensure that our students are aware of and can make use of modern techniques of system implementation. Sommerville suggests that all teaching courses should have some material introducing students to cloud computing and the practical teaching should focus on Platform as a Service. A graduate course in cloud software engineering, for example, should cover map-reduce, schema-free databases, service-oriented computing, security and compliance and design for resilience among the other topics.

A special issue can only hope to represent a small selection of current research. Nevertheless, this issue has collected an interesting cross-section of papers that contribute to the state of the art, and that we hope will stimulate further work in the area of cloud software engineering. We would like to thank all the authors who contributed and the many referees, who played an important part in the selection process. Thanks for the editorial and publication team for their helpful support during the process. Last but not least, a special thank is due to the Editor-in-Chief of JSS, Hans van Vliet, whose enthusiastic support, careful eye on details and immense devotion in editorial duty and service enabled the Special Issue to proceed and flourish from his continuous constructive feedback and comments.

About the Guest-Editors:

Rami Bahsoon is a Senior Lecturer (equivalent to Associate Professor) in Software Engineering and founder of the Cloud Software Engineering Interest Group at the University of Birmingham, UK where he leads a group of PhD students and postdoctoral researchers conducting research in cloud software architecture, cloud requirements engineering and economics-driven software architecting for the cloud. His current research aims at developing economics-driven frameworks to support and reason about the development and evolution of *dependable* complex software systems such as cloud-based systems. He is a founding member of the IEEE International Software Engineering IN/FOR the Cloud workshop in conjunction with IEEE Cloud and IEEE Services; the premier cloud computing meeting. He had co-organised the Architectural and Mobility Workshop in conjunction with ICSE and had guest edited an issue on the theme with JSS. His contributions, include publications in leading conferences and journals on software architecture and cloud software engineering (e.g., Transactions on Software Engineering, International Conference on Software Architecture (WICSA), SEAMS etc.). He is regular reviewer for premier software engineering journals, conferences, and funding bodies including NSF and EPSRC. He had acted as a panellist for the NSF/EPSRC Software Grand Challenge for his expertise in cloud software engineering. He is currently co-editing two books on software architectures. Rami holds a PhD in Software Engineering from University College London for his research on economics-driven software architectures. He had also read for MBA in technology strategy and new communications industry at London Business School.

Ivan Mistrik is an independent consultant and researcher in software-reliant systems engineering. He is a computer scientist who is interested in software engineering (SE) and software architecture (SA), in particular: life cycle software engineering, requirements engineering, aligning

enterprise/system/software architectures, knowledge management in software development, global software development, and collaborative software engineering. He has more than forty years' experience in the field of computer systems engineering as an information systems developer, R&D leader, SE/SA research analyst, educator in computer sciences, and ICT management consultant. In the past 40 years, he has been primarily working at R&D institutions in USA and Germany and has done consulting on a variety of large international projects sponsored by ESA, EU, NASA, NATO, and UN. He has also taught university-level computer sciences courses in software engineering, software architecture, distributed information systems, and human-computer interaction. He is the author or co-author of more than 80 articles and papers in international journals, conferences, books and workshops, most recently a chapter Capture of Software Requirements and Rationale through Collaborative Software Development, a paper Knowledge Management in the Global Software Engineering Environment, and a paper Architectural Knowledge Management in Global Software Development. He has also written over 90 technical reports and presented over 70 scientific/technical talks. He has served in many program committees and panels of reputable international conferences and organized a number of scientific workshops, most recently two workshops on Knowledge Engineering in Global Software Development at International Conferences on Global Software Engineering in 2009 and in 2010 (proceedings of these workshops have been published by IEEE Computer Society, Conference Publishing Services). He has been a guest-editor of IEE Proceedings Software: A special Issue on Relating Software Requirements and Architectures published by IEE in 2005 and a lead-editor of the book Rationale Management in Software Engineering published by Springer in 2006. He has been a co-author of the book Rationale-Based Software Engineering published by Springer in May 2008. He has been a lead-editor of the book Collaborative Software Engineering published in 2010. Currently he is an editor of the book on Relating Software Requirements and Architectures and a lead editor of the Expert Systems Special Issue on Knowledge Engineering in Global Software Development and another two books on software architectures.

Nour Ali is a senior lecturer in software engineering at the University of Brighton. Prior to that, she was a Research Fellow at Lero-The Irish Software Engineering Research Centre, Ireland. She graduated in Computer Science from Bir-Zeit University, Palestine and holds a PhD. in Software Engineering and Declarative Languages from the Polytechnic University of Valencia, Spain. During her PhD, she was a visiting researcher at University of Leicester, UK. Her PhD. is on specifying software architecture of distributed and mobile systems using model driven engineering techniques for generating and executing their code. She has been an active member of several research projects focusing on software architecture such as the Microsoft Research Cambridge funded Project: "PRISMA Model Compiler of aspect-oriented component-based software architectures". She is on the editorial board of the Int. Journal of Software Architectures. She also serves as reviewer for several venues and journals such as Journal of Systems and Software, Journal of Information Systems and Technology, and IEEE Transactions on Systems, Man, and Cybernetics. She has co-chaired and organized the IEEE International Workshop on Engineering Mobile Service Oriented Systems (EMSOS). Her research interests are software architecture, service oriented engineering, distributed and mobile systems, aspect-oriented software development, model driven engineering and global software development.

T.S. Mohan works at Infosys Technologies E&R's ECom Research Lab as a Principal Researcher. His research interests include distributed systems, high performance computing, cloud and grid as well

as software architecture and Software engineering. He has over 24 years experience in the academia and industry. T.S. Mohan holds a Master and PhD in computer science from the Indian Institute of Science, Bangalore where he worked for about a decade before moving into the industry. He was a young visiting scientist in the Lab for Computer Science, MIT in 1988 and a visiting scientist in NEC Research Institute, Princeton in the summer of 1994. He pursued his entrepreneurial interests in Bangalore in advanced computing technologies for about 6 years before joining Infosys.

Nenad Medvidović is a Professor and Associate Chair for Ph.D. Affairs in the Computer Science Department at the University of Southern California. Between 2009 and 2013 Medvidović served as Director of the USC Center for Systems and Software Engineering (CSSE). He is a faculty associate of the Institute for Software Research (ISR) at the University of California, Irvine. Medvidović was the Program Co-Chair of the 2011 International Conference on Software Engineering (ICSE 2011). Medvidović received his Ph.D. in 1999 from the Department of Information and Computer Science at UC Irvine. He also received an M.S. in Information and Computer Science in 1995 from UC Irvine, and a B.S. in Computer Science summa cum laude in 1992 from the Computer Science and Engineering Department at Arizona State University. Medvidović is a recipient of the National Science Foundation CAREER (2000) award, the Okawa Foundation Research Grant (2005), the IBM Real-Time Innovation Award (2007), and the USC Mellon Mentoring Award (2010). He is a co-author of the ICSE 1998 paper titled “Architecture-Based Runtime Software Evolution”, which was recognized as that conference’s Most Influential Paper. His paper “A Classification and Comparison Framework for Software Architecture Description Languages” was recognized by the Elsevier Information and Software Technology Journal as the most cited journal article in software engineering published in 2000. Medvidović’s research interests are in the area of architecture-based software development. His work focuses on software architecture modeling and analysis; middleware facilities for architectural implementation; domain-specific architectures; architectural styles; and architecture-level support for software development in highly distributed, mobile, resource constrained, and embedded computing environments. He is a co-author of a textbook on software architectures. Medvidović is a member of ACM, ACM SIGSOFT, IEEE, and IEEE Computer Society.

â€¢ The future of cloud computing: Why CFD engineers should move to cloud-based solutions. What Is SaaS? Software as a service (SaaS) is a software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted.Â

Predictions for the Future of Cloud Computing and Cloud-Based CFD. Iâ€™m sure weâ€™ll see an increase in CAD complexity, and other CAE software like Onshape will be ready for this change. I think it can be predicted that AWS will continue to be the front runner for cloud services, but with a greater focus on security. My musings on why it might be the time to start looking for a new profession if you are a software engineer? You know how in the early 20th century pilots were like heroes? It was extremely dangerous to fly planes which were highly unreliable.Â Who do you think will DEVELOP the future Excels and the future drag and drop environments? People who write Excel macros and drag controls to a form? No. Programmers will. Maybe the only problem is the name â€” go ahead and call those macro-scribblers â€œsoftware engineersâ€, and those of us who develop the programs they have to use will just take a new name. BTW, airline pilots make ONE HECK OF A LOT MORE MONEY than bus drivers. Is cloud computing really the future for all industries or are there businesses that would be better off keeping their systems on-premise? We believe in a future that benefits from both on-premise and cloud resources. There is a need for both and different industriesâ€”and certainly different organizationsâ€”will have distinct needs and considerations influencing the degree to which these options are tapped.Â Computer scientists and other technology purists are the answer to sustaining job growth in IT now and in the future. However, we must not lose sight of the need for platforms that ultimately overcome operational complexity, reduce the learning curve, and allow non-IT users to run their workloads without needing IT skills themselves. (Image credit: Image Credit: RawPixel / Unsplash). This means that the future of software development is actually happening now. And the future is present in the current trends being adopted by software development teams.

Current Trends in Software Development. There are many trends being embraced today that will drive the future of software development. But where the trends are at â€” in terms of adoption â€” varies between innovators who are ahead of the curve, early adopters who are next up, and early majority who are embracing the trends. Innovators are adopting: Deep learning: a subset of machine learning based on artificial neural networks.Â Rod holds both Bachelor's and Master's degrees in Software Engineering from the University of Louisville. Recommended Posts. January 22, 2021. Service-oriented software engineering incorporates the best features of both the services and cloud computing paradigms, offering many advantages for software development and applications, but also exacerbating old concerns.Â Some researchers envision the future Internet as a kind of supercomputer that will depend heavily on cloud computing features, such as resource pooling and virtualization, on-demand service, and ubiquitous access. There are four cloud types. A public cloud provides services to the public.