Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
Editorial

Introduction to the special issue on managing software processes using soft computing techniques

ABSTRACT

The coronavirus outbreak dramatically changed the work culture in the software industry. Most software practitioners began working remotely, which significantly revolutionized the traditional software processes landscape. Software development organizations have begun thinking about automating software processes to cope with the challenges raised by remote work. This special issue presents papers describing soft computing solutions for improving traditional software processes and capabilities. This editorial introduces the accepted papers and reflects on their contributions.

1. Introduction

The COVID-19 outbreak substantially impacts a vast array of industries, specifically the global tech sector. The leading countries (e.g., USA) of the global tech supply chain are greatly affected by the pandemic, which caused limited operations of many technical units. Various industries got a sudden jolt because of the COVID 19 pandemic. However, the software sector felt significant tremors due to the intensive interconnected market. The work culture in the software development industry is highly affected by the outbreak and has been continuously reshaped over the last couple of years. More workplaces open by scaling the development activities across remote locations, which drives significant changes in modern-day collaborative, distributed, opaque, and dynamic software processes. For instance, the continuous software engineering practices (e.g., automation, continuous delivery, continuous integration, infrastructure-as-code) used in industrial settings are tremendously evolving due to the pandemic and researchers are making innovative efforts to cope with the challenges. The increasing demand for continuous software development, the use of iterative practices, and the development of software in a globally distributed environment have become the real industrial challenges. Software development firms are struggling to follow continuous and agile software development practices in distributed environments, which cannot be achieved using the traditional process models e.g., waterfall, and spiral.

Software teams require various intelligent and soft computing techniques to manage these software process challenges. Soft computing is a collection of computational methods that focus on exploiting the tolerance of imprecision to realize robustness, low solution cost, and tractability. Soft computing plays a significant role in providing cost-effective solutions to complex real-world science and engineering problems. Soft computing is broadly applicable to topics including wireless communication, consumer appliances, transportation, healthcare, aerospace, automobile, and power engineering. Similarly, soft computing has a list of applications across software engineering life cycle phases, e.g., software testing, bug triaging, matrices, estimation, debugging, and formal methods. However, the implications of soft computing techniques across software processes are still unexplored. It seems more appropriate field because of incorporating people, tools, and techniques to develop an automated process flow.

Further investigation of soft computing techniques and their suitability in the software process improvement and information processing domain is required. The applied soft computing methods, e.g., probabilistic reasoning, fuzzy decision making, evolutionary computation, swarm intelligence, machine learning, and bayesian networks, seem to be a natural fit for developing a roadmap for managing and automating software processes. Computationally mature processes and capabilities could be a quantum leap in automating software development activities.

This special issue draws the attention of researchers and practitioners to technical strategies and empirical evidence about those strategies as well as to encourage future studies of soft computing applications in software processes. The special issue scope is not limited to specific soft computing methods but includes the broad-level applications of soft computing techniques to propose new tools, models, standards, practices, and frameworks to maximize the significance of software processes.

2. Review process

We received 21 submissions in response to the call for papers. Of those, we desk rejected seven because they were out of scope for the special issue. Each of the remaining 14 manuscripts were reviewed by two reviewers. After a process of major and minor revisions, we finally accepted seven because they were out of scope for the special issue. Each of the remaining 14 manuscripts were reviewed by two reviewers. After a process of major and minor revisions, we finally accepted seven manuscripts for publication in this special issue.

3. Papers accepted in this special issue

In their article “Toward successful DevSecOps in software development organizations: A decision-making framework,” Muhammad
Azeem Akbar, Kari Smolaner, Sajjad Mahmood, and Ahmed Alsanad propose a taxonomy of challenging factors in DevSecOps through a multivocal literature review (MLR) and a questionnaire-based survey study. After identifying the factors through the MLR, the questionnaire-based survey validated the findings and identified additional challenges not reported in the literature. They mapped these factors across ten core categories and present a taxonomy. Finally, they applied the fuzzy TOPSIS soft computing approach to prioritize the challenges and categories of the proposed taxonomy. The fuzzy TOPSIS findings reveal the most significant DevSecOps challenges are: lack of secure coding standards, lack of automated testing tools for security in DevOps, and ignorance in static testing for security due to lack of knowledge.

In the second article, "Quantum computing challenges in the software industry. A fuzzy AHP-based approach" Usama Awan, Lea Hananola, Anushree Tandon, Raman Kumar Goyal, and Amandeep Dhir present another taxonomy. In contrast with the taxonomy in the first article, these authors use fuzzy AHP soft computing techniques to prioritize the challenges of quantum computing in the software industry. The authors review a systematic literature review to identify and categorize the relevant challenges. They then use the fuzzy AHP technique to prioritize the reported challenges and their categories. The findings provide a framework of quantum computing challenges that assist in avoiding critical barriers before scaling the software development activities across the quantum computing domain.

The third article, "Predicting reliability of software in industrial systems using a Petri net based approach: A case study on a safety system used in nuclear power plant" by Kuldeep Kumar, Sumit, Sandeep Kumar, Lalit Kumar Singh, and Alok Mishra presents a robust framework for predicting industrial software systems reliability using a Petri net based approach. They evaluate the significance of the proposed framework through an industrial case study of a safety-critical system in a Canadian nuclear power plant. The case study found the accuracy of the proposed approach is 99.9%, which is relatively high compared to the system specifications.

The fourth article, "Prioritization of model smell refactoring using a covariance matrix-based adaptive evolution algorithm" by Amjad AbuHassan, Mohammad Alshayeb and Lahouari Ghouti proposes a novel approach for model smell refactoring using a multi-objective covariance matrix adaptation evolution strategy (MO-CMA-ES) algorithm. They evaluate the approach with a large custom dataset containing more than 30,000 class records. The evaluation results revealed that the proposed approach effectively fixed the identified design smells and considered it best for improving software quality and maintainability.

The fifth article, "An end-to-end deep learning system for requirements classification using recurrent neural networks" by Osamah AlDhafer, Irfan Ahmad, and Sajjad Mahmood proposes a framework to classify functional and non-functional software requirements using Bidirectional Gated Recurrent Neural Networks (BiGRU). The research motivation is explicitly based on the existing frameworks’ binary or multiclass requirements classification problems. The authors mentioned that a given set of requirements could belong to multiple classes simultaneously. They use the approach to classify reports based on the publicly available ROMISE dataset. The results confirm significant improvements in the requirements classification process.

The sixth article, "Undulate: A framework for data-driven software engineering enabling soft computing" by Timo Asikainen, and Tomi Mannist, presents the Undulate framework for automating the management of usage data from software and business processes. The authors explore the literature with a multilevel modelling language to process, augment, and aggregate the usage data. The proposed framework enables the application of soft computing and AI-based methods in presenting a roadmap for standardizing the processes in the data-driven software engineering domain. In the long term, the authors describe the need for in-practice evaluation of the proposed framework to understand its real-world implications.

The seventh article "Taxonomy of Bug Tracking Process Smells: Perceptions of Practitioners and an Empirical Analysis" by Khushbakht Ali Qamar, Emre Sülün and Eray Tüzün, proposes a taxonomy of smells in the bug tracking process by adopting three different approaches: MLR, repositories mining, and survey questionnaire. The MLR explores the literature to develop the preliminary taxonomy of the bug tracking process smells. In the next phase, the authors use an empirical study to evaluate the MLR-based smells by mining the bug reports of eight open source projects available at Jira, Bugzilla, and GitHub. Finally, the authors conduct a survey questionnaire to encapsulate practitioners’ perceptions regarding the taxonomy of the identified smells. Statistical analysis provides insights into the impacts of the reported smells on software quality and speed of bug recovery. In conclusion, the authors considered the proposed taxonomy a foundational tool to support bug tracking process activities of detecting and avoiding smells.

4. Conclusions

The articles in this special issue offer different approaches for soft computing applications in software processes. We hope that the findings of these articles will inspire and encourage future research studies focusing on soft computing processes. Besides the proposed taxonomies, the papers present several robust frameworks to automate process activities. Together, the proposed taxonomies and frameworks offer various future opportunities to investigate how the extant soft computing techniques could use to tackle the emerging software process challenges, for example:

- Evaluate the proposed taxonomies and frameworks through further empirical studies for generalizing their implications across a vast array of processes and gauge if they explicitly captured the critical concerns of the software industry.
- Comparative analysis of the proposed taxonomies and frameworks to know the best fit for explaining applications of soft computing techniques in the software process domain.
- Develop novel automated tools, frameworks, models, and standards based on knowledge and solution-seeking studies for quantum software development processes.

We believe that further research on how the soft computing applications for automating advanced and state-of-art process activities can contribute to process-centric software engineering research is warranted.

Acknowledgments

We would like to thank all the authors who submitted to the special issue. We also thank all the reviewers for their time and effort in reviewing the assigned articles and providing valuable comments and suggestions for improving the quality of the submitted studies. Last but not least, we are grateful to Prof. Jeffrey Carver, the Information and Software Technology Journal special issue editor, for continuous assistance and guidance in managing the special issue activities.
Arif Ali Khan works as an Assistant Professor with the M3S Empirical Software Engineering Research Unit, University of Oulu, Finland. Previously he worked as a faculty member with the Faculty of Information Technology, University of Jyväskylä, Finland; College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China; and Department of Computer Science, COMSATS University, Islamabad, Pakistan. He obtained a PhD degree in software engineering from the Department of Computer Science, City University of Hong Kong. He has participated in and managed several empirical software engineering-related research projects. Khan has expertise in software process improvement, global software development, multi-criteria decision analysis, DevOps, quantum software architecture, microservices architecture, AI ethics, agile software development, requirements change management, soft computing, and evidence-based software engineering. He is professionally active in conducting publication-based research workshops, serving as guest editor in main track software engineering journals, and editing software engineering research books. He has published over 80 articles in peer-reviewed software engineering conferences and journals.

Pekka Abrahamsson works as a full professor of information systems and software engineering at the University of Jyväskylä in Finland. He received his PhD on Software Engineering in 2002 from University of Oulu. His research is in the area of emerging software technologies, empirical software engineering, software startups, and the ethics of artificial intelligence. Before his current position, he has served as full professor in University of Helsinki (Finland), Free University of Bolzano (Italy), Norwegian University of Science and Technology (Norway). He also worked at VTT Technical Research Centre of Finland as a research professor of software technologies. He is widely recognized for his academic achievements. He is a pioneer in the field of research on agile software engineering methods and processes. Abrahamsson is the most cited researcher in his field in Finland, and he is the first Professor of Software Engineering at the Finnish Academy of Science and Letters. He has published broadly in his areas of expertise and received many awards and recognitions. Arnetminer named him amongst the 100 most influential software engineering scientists in the world in 2016. Abrahamsson was awarded the Nokia Foundation Award 2007. In the same year, his large international European research project was also awarded the ITEA-Achievement Award. He is the co-founder of the Software Startup Research Network (SSRN) and a seasoned expert in leading large research projects.

Mahmood Niazi received the M.Phil. degree from the University of Manchester, U.K., and the Ph.D. degree from the University of Technology Sydney Australia. He is currently a full professor with the Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Saudi Arabia. He is also an active Researcher in the field of software engineering and have published more than 100 papers in peer-reviewed conferences and journals. He has won many research grants, such as EPSRC and European Commission. He also worked as an editor for various journals and conferences.

Arif Ali Khan\textsuperscript{a*}, Pekka Abrahamsson\textsuperscript{b}, Mahmood Niazi\textsuperscript{c}
\textsuperscript{a} M3S Empirical Software Engineering Research Unit, University of Oulu, 90570 Oulu, Finland
\textsuperscript{b} University of Jyväskylä, Jyväskylä, Finland
\textsuperscript{c} Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Saudi Arabia

* Corresponding author.

E-mail address: arif.khan@oulu.fi (A.A. Khan).