Postprint of article in *Software: Practice and Experience*

Editorial:
Focus Section on Quality Software

Developing software systems to fulfill the requirements of various stakeholders is by no means a simple matter. Quality assurance is required in each phase of the software engineering process including requirements elicitation, software architecture design, program design, implementation, testing, and debugging, since every phase is closely linked with another. The quality of the artifacts from each development phase impacts on the rest of the system. The international conference series on quality software has a long tradition of bringing together researchers and practitioners to present and discuss innovative methods of assuring software quality.

The 13th International Conference on Quality Software (QSIC 2013) was held in Nanjing, China on July 29–30, 2013. The main theme was on the quality of evolving software. We emphasized a holistic view of quality assurance across different phases and aspects of software engineering. QSIC 2013 was technically sponsored by the IEEE Reliability Society. Jian Lv was the General Chair. Arnaud Gotlieb and Zhenyu Chen served as the Program Chairs. The keynote speakers were Mauro Pezzè of Università della Svizzera Italiana, Switzerland and Magne Jorgensen of Simula Research Laboratory, Norway. 78 submissions from 21 countries were received. 19 regular papers were accepted, representing an acceptance rate of 24%. We had an industry track where experience reports from practitioners were presented. Roberto Bagnara of University of Parma, Italy, cofounder of BUGSENG, was the invited industry speaker. In addition, The Symposium on Engineering Test Harness (TSETH 2013), the Workshop on Testing and Verification of Embedded Computing Systems (TVECS 2013), the Workshop on Quality and Measurement of Software Model-Driven Developments (QUAMES 2013), and the Workshop on Software Quality Assurance of Healthcare System and Embedded System (SQHE 2013) were also held. The Proceedings of QSIC 2013 was published by IEEE Computer Society.

We shortlisted six papers from the main conference and invited the authors to submit extended versions to this Focus Section on Quality Software in *Software: Practice and Experience*. Two papers were accepted after going through up to three rounds of rigorous reviews involving two anonymous reviewers for each article.

Automated tools are essential for every stage of the system development life cycle to support the computer-aided software engineering process. There is an abundance of tools to be selected for the different phases. Comparing their effectiveness and the ability to integrate with one another is a nontrivial task. The first paper, entitled “Selecting a Software Engineering Tool: Lessons Learnt from Mutation Analysis” by Mickaël Delahaye and Lydie du Bousquet, studies the comparison and choice of mutation analysis tools as an illustration of their proposed methodology for tool selection. Mutation analysis involves the seeding of faults into programs under test and verifies whether the test suites can detect such faults. Mutation tools vary in the fault models used and their performance in regard to such issues as fault generation and test suite execution. The authors propose a list of comparison criteria for such tools and a list of usage profiles. They find the listing of criteria to be straightforward but their appraisals to be much harder. They have evaluated the mutation tools for the Java platform. Generalizations to other platforms and other tools are also discussed. This paper is of interest to software testers working on mutation analysis as well as software developers who need to choose which automated tools to use.

Safety requirements are crucial to every development phase of an avionic system. The second paper, entitled “A Modeling Methodology to Facilitate Safety-Oriented Architecture Design of Industrial Avionics Software” by Ji Wu, Tao Yue, Shaukat Ali, and Huihui Zhang, presents a Safety-Oriented Architecture Modeling (SOAM) methodology to enforce adherence of the avionic system under development to published standards and industrial practices. The authors propose a UML profile to define the safety requirements in terms of a component-based architecture, a modeling environment to assure the implementation of such requirements, and design guidelines including objectives and processes for applying the model. The safety requirements are based on the DO-178B/C standard as well as a systematic domain analysis of current engineering practices. To evaluate the methodology, it has been applied to an industrial autopilot system. All the stereotypes in the safety profile have been verified. 32 safety properties have been identified and are checked between the formal UML profile and the architectural model. Six faults previously unrevealed have been identified. This paper should be of interest not only to developers of avionics software but also serve as a good reference to others who are concerned about safety-critical systems.

Finally, we would like to thank the editors of *Software: Practice and Experience* for kindly agreeing to publish this focus section.

T.H. TSE
Department of Computer Science
The University of Hong Kong
Pokfulam, Hong Kong

ARNAUD GOTLIEB
Simula Research Laboratory
Norway

ZHENYU CHEN
Software Institute
Nanjing University
China