Special Issue of Software Quality Journal
on
Program Debugging: Research, Practice and Challenges

Guest Editors:
Sudipto Ghosh – Colorado State University - ghosh@CS.ColoState.EDU

Jenny Li – Avaya Lavs -- juli@kean.edu

Background

Software today is large and complex, in fact more so than ever before. Consequently, debugging when failure is observed is also becoming much more difficult and time-consuming. Manual debugging is quickly losing its viability as a practical option, and yet at the same time, various alternative approaches may still be too immature to use practically. Techniques that aim for automatic fault localization are not accurate and consistent enough to pinpoint the locations of faults to a desired degree. Distinguishing executions that fail due to different causative faults, reliably recording and replaying failed executions, and fixing bugs without introducing new faults are but some of the debugging-related problems faced by developers today. Recent efforts such as the recommender systems approaches that mine different types of software repositories and suggest various debugging actions or program fixes are still unproven to be consistently effective. Formal verification techniques generally suffer from complexity and scalability issues, static techniques can often be imprecise, and the heavy performance overhead of dynamic techniques can prohibit their application. Even though a software development project may apply such techniques, they may require developers to make final selection, diagnosis, and fixing decisions. Social aspects of software development projects that aid debugging, such as selecting the right developers to perform the right debugging tasks at the right time, have not been adequately explored. Last but not least, while studies are being conducted to reveal, clarify, or resolve some of these issues, researchers often conduct studies in restrictive environments that may inherently make unwelcome assumptions about the industry. All these concerns can induce in practitioners a lack of faith with regard to the results that debugging research can offer and deliver.

Topics

The topics of interest include, but are not limited to:

- Automation of software debugging activities
- Cost-effective approaches to debugging large scale applications
- Challenges and emerging techniques in program debugging for large scale real-life applications and domain-specific applications
- Debugging for multi-(core, process, or threaded) programs
- Empirical studies and open source-based benchmarking infrastructure
- Experience reports and industrial best practices
- Impacts of software business, human factors, programming languages, and tool environments on program debugging
- Integrating debugging with other software development and maintenance activities
- Social aspects of program debugging

- Software risk analysis and cost estimation for fault localization, bug fixing, and their social interactions
- New frontiers on mining software repositories for debugging
- Effective software risk analysis for software debugging
- New frontiers on fault proneness prediction for software debugging
- Empirical studies on debugging activities and/or techniques, or bug repositories
- Large-scale software debugging benchmarks
- Transitioning from research to practice
- Effective pedagogical models to teaching software debugging

<u>Submission Information</u>

There are two types of submissions to this special issue. The first category includes the best papers relevant to the above topics selected from the Fifth International Workshop on Program Debugging (IWPD 2014; http://paris.utd allas.edu/iwpd14), collocated with the 25th IEEE International Symposium on Software Reliability Engineering (ISSRE) which will be held in Naples, Italy in November 2014. Authors of selected papers are invited to submit an extended version with at least 30% difference in technical content. Second, we also solicit papers from the research community with an open call for papers. Every submission must be of high quality and original, not published or currently submitted elsewhere. Every submission will also be evaluated by at least three independent reviewers, using the same review process and standard for SQJ regular submissions.

The Editor-in-Chief, along with the guest editors, will make the final decision to accept or decline a submission based on the reviews. Submissions must be written in English and submitted in the PDF format via the Editorial Manager's system at https://www.editorialmanager.com/sqjo/