# Contents

## Preface

Message from the Chairs ........................................ xix
Committees ....................................................... xxiii
Sponsors .......................................................... xxx

## Technical Research

### Fault Handling

A Systematic Study of Automated Program Repair: Fixing 55 out of 105 Bugs for $8 Each  
Claire Le Goues, Michael Dewey-Vogt, Stephanie Forrest, and Westley Weimer — University of Virginia, USA; University of New Mexico, USA ........................................ 3

Where Should the Bugs Be Fixed? - More Accurate Information Retrieval-Based Bug Localization Based on Bug Reports  
Jian Zhou, Hongyu Zhang, and David Lo — Tsinghua University, China; Singapore Management University, Singapore .................................................. 14

Developer Prioritization in Bug Repositories  
Jifeng Xuan, He Jiang, Zhilei Ren, and Weiqin Zou — Dalian University of Technology, China .................................................. 25

WhoseFault: Automatic Developer-to-Fault Assignment through Fault Localization  
Francisco Servant and James A. Jones — UC Irvine, USA .................................................. 36

### Code Generation and Recovery

Recovering Traceability Links between an API and Its Learning Resources  
Barthélémy Dagenais and Martin P. Robillard — McGill University, Canada .................................................. 47

Generating Range Fixes for Software Configuration  
Yingfei Xiong, Arnaud Hubaux, Steven She, and Krzysztof Czarnecki — University of Waterloo, Canada; University of Namur, Belgium .................................................. 58

Graph-Based Pattern-Oriented, Context-Sensitive Source Code Completion  
Anh Tuan Nguyen, Tung Thanh Nguyen, Hoan Anh Nguyen, Ahmed Tamrawi, Hung Viet Nguyen, Jafar Al-Kofahi, and Tien N. Nguyen — Iowa State University, USA .................................................. 69

Automatic Input Rectification  
Fan Long, Vijay Ganesh, Michael Carbin, Stelios Sidiroglou, and Martin Rinard — MIT, USA .................................................. 80

### Empirical Studies of Development

Overcoming the Challenges in Cost Estimation for Distributed Software Projects  
Narayan Ramasubbu and Rajesh Krishna Balan — Singapore Management University, Singapore .................................................. 91

Characterizing Logging Practices in Open-Source Software  
Ding Yuan, Soyeon Park, and Yuanyuan Zhou — University of Illinois at Urbana-Champaign, USA; UC San Diego, USA .................................................. 102

The Impacts of Software Process Improvement on Developers: A Systematic Review  
Mathieu Lavallée and Pierre N. Robillard — École Polytechnique de Montréal, Canada .................................................. 113
Combining Functional and Imperative Programming for Multicore Software: An Empirical Study Evaluating Scala and Java
Victor Pankratius, Felix Schmidt, and Gilda Garretón — KIT, Germany; Oracle Labs, USA 123

Performance Analysis
Uncovering Performance Problems in Java Applications with Reference Propagation Profiling
Dacong Yan, Guoqing Xu, and Atanas Rountev — Ohio State University, USA; UC Irvine, USA 134

Performance Debugging in the Large via Mining Millions of Stack Traces
Shi Han, Yingnong Dang, Song Ge, Dongmei Zhang, and Tao Xie — Microsoft Research, China; North Carolina State University, USA 145

Automatically Finding Performance Problems with Feedback-Directed Learning Software Testing
Mark Grechanik, Chen Fu, and Qing Xie — Accenture Technology Labs, USA; University of Illinois at Chicago, USA 156

Predicting Performance via Automated Feature-Interaction Detection
Norbert Siegmund, Sergiy S. Kolesnikov, Christian Kästner, Sven Apel, Don Batory, Marko Rosenmüller, and Gunter Saske — University of Magdeburg, Germany; University of Passau, Germany; Philipps University of Marburg, Germany; University of Texas at Austin, USA 167

Defect Prediction
Sound Empirical Evidence in Software Testing
Gordon Fraser and Andrea Arcuri — Saarland University, Germany; Simula Research Laboratory, Norway 178

Privacy and Utility for Defect Prediction: Experiments with MORPH
Fayola Peters and Tim Menzies — West Virginia University, USA 189

Bug Prediction Based on Fine-Grained Module Histories
Hideaki Hata, Osamu Mizano, and Tohru Kikuno — Osaka University, Japan; Kyoto Institute of Technology, Japan 200

Refactoring
Reconciling Manual and Automatic Refactoring
Xi Ge, Quinton L. DuBose, and Emerson Murphy-Hill — North Carolina State University, USA 211

WitchDoctor: IDE Support for Real-Time Auto-Completion of Refactorings
Stephen R. Foster, William G. Griswold, and Sorin Lerner — UC San Diego, USA 222

Use, Disuse, and Misuse of Automated Refactorings
Mohsen Vakilian, Nicholas Chen, Stas Negara, Balaji Ambresh Rajkumar, Brian P. Bailey, and Ralph E. Johnson — University of Illinois at Urbana-Champaign, USA 233

Human Aspects of Development
Test Confessions: A Study of Testing Practices for Plug-In Systems
Michaela Greiler, Arie van Deursen, and Margaret-Anne Storey — TU Delft, Netherlands; University of Victoria, Canada 244

How Do Professional Developers Comprehend Software?
Tobias Roehm, Rebecca Tiarks, Rainer Koschke, and Walid Maalej — TU Munich, Germany; University of Bremen, Germany 255

Asking and Answering Questions about Unfamiliar APIs: An Exploratory Study
Ekwa Duala-Ekoko and Martin P. Robillard — McGill University, Canada 266

Bug Detection
Automated Repair of HTML Generation Errors in PHP Applications Using String Constraint Solving
Hesam Samimi, Max Schäfer, Shay Artzi, Todd Millstein, Frank Tip, and Laurie Hendren — UC Los Angeles, USA; IBM Research, USA; McGill University, Canada 277

Leveraging Test Generation and Specification Mining for Automated Bug Detection without False Positives
Michael Pradel and Thomas R. Gross — ETH Zurich, Switzerland 288

Axis: Automatically Fixing Atomicity Violations through Solving Control Constraints
Peng Liu and Charles Zhang — Hong Kong University of Science and Technology, China 299

CBCD: Cloned Buggy Code Detector
Jingyue Li and Michael D. Ernst — DNV Research and Innovation, Norway; University of Washington, USA 310
Multiversion Software
Crosscutting Revision Control System
Sagi Ifrah and David H. Lorenz — Open University, Israel

Where Does This Code Come from and Where Does It Go? - Integrated Code History Tracker for Open Source Systems -
Katsuro Inoue, Yusuke Sasaki, Pei Xia, and Yuki Manabe — Osaka University, Japan

Improving Early Detection of Software Merge Conflicts
Mário Luís Guimarães and António Rito Silva — Technical University of Lisbon, Portugal

A History-Based Matching Approach to Identification of Framework Evolution
Sichen Meng, Xiaoyin Wang, Lu Zhang, and Hong Mei — Key Laboratory of High Confidence Software Technologies, China; Feking University, China

Similarity and Classification
Detecting Similar Software Applications
Collin McMillan, Mark Grechanik, and Denys Poshvanyk — College of William and Mary, USA; Accenture Technology Labs, USA; University of Illinois at Chicago, USA

Content Classification of Development Emails
Alberto Bacchelli, Tommaso Dal Sasso, Marco D’Ambros, and Michele Lanza — University of Lugano, Switzerland

Identifying Linux Bug Fixing Patches
Yuan Tian, Julia Lawall, and David Lo — Singapore Management University, Singapore; INRIA/LIP6, France

Active Refinement of Clone Anomaly Reports
Lucia, David Lo, Lingxiao Jiang, and Aditya Budi — Singapore Management University, Singapore

Analysis for Evolution
Automated Analysis of CSS Rules to Support Style Maintenance
Ali Mesbah and Shabnam Mirshokraie — University of British Columbia, Canada

Graph-Based Analysis and Prediction for Software Evolution
Pamela Bhattacharya, Marios Iliofotou, Iulian Neamtiu, and Michalis Faloutsos — UC Riverside, USA

Integrated Impact Analysis for Managing Software Changes
Malcom Gethers, Bogdan Dit, Huzefa Kagdi, and Denys Poshvanyk — College of William and Mary, USA; Wichita State University, USA

Detecting and Visualizing Inter-worksheet Smells in Spreadsheets
Felienne Hermans, Martin Pinzger, and Arie van Deursen — TU Delft, Netherlands

Debugging
An Empirical Study about the Effectiveness of Debugging When Random Test Cases Are Used
Mariano Ceccato, Alessandro Marchetto, Leonardo Mariani, Cu D. Nguyen, and Paolo Tonella — Fondazione Bruno Kessler, Italy; University of Milano-Bicocca, Italy

Reducing Confounding Bias in Predicate-Level Statistical Debugging Metrics
Ross Gore and Paul F. Reynolds, Jr. — University of Virginia, USA

BugRedux: Reproducing Field Failures for In-House Debugging
Wei Jin and Alessandro Orso — Georgia Tech, USA

Object-Centric Debugging
Jorge Rescia, Alexandre Bergel, and Oscar Nierstrasz — University of Bern, Switzerland; University of Chile, Chile

Human Aspects of Process
Disengagement in Pair Programming: Does It Matter?
Laura Plonka, Helen Sharp, and Janet van der Linden — Open University, UK

Ambient Awareness of Build Status in Collocated Software Teams
John Downs, Beryl Plimmer, and John G. Hosking — University of Melbourne, Australia; University of Auckland, New Zealand; Australian National University, Australia
<table>
<thead>
<tr>
<th>Title</th>
<th>Authors</th>
<th>Location(s)</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>What Make Long Term Contributors: Willingness and Opportunity in OSS Community</td>
<td>Minghui Zhou and Audris Mockus</td>
<td>Peking University, China; Key Laboratory of High Confidence Software Technologies, China; Avaya Labs Research, USA</td>
<td>518</td>
</tr>
<tr>
<td>Development of Auxiliary Functions: Should You Be Agile? An Empirical Assessment of Pair Programming and Test-First Programming</td>
<td>Otávio Augusto Lazzarini Lemos, Fabiano Cutugi Ferrari, Fábio Fagundes Silveira, and Alessandro Garcia</td>
<td>UNIFESP, Brazil; UFSCar, Brazil; PUC-Rio, Brazil</td>
<td>529</td>
</tr>
<tr>
<td>Models</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintaining Invariant Traceability through Bidirectional Transformations</td>
<td>Yijun Yu, Yu Lin, Zhenjiang Hu, Soichiro Hidaka, Hiroyuki Kato, and Lionel Montreux</td>
<td>Open University, UK; University of Illinois at Urbana-Champaign, USA; National Institute of Informatics, Japan</td>
<td>540</td>
</tr>
<tr>
<td>Slicing MATLAB Simulink Models</td>
<td>Robert Reicherdt and Sabine Glesner</td>
<td>TU Berlin, Germany</td>
<td>551</td>
</tr>
<tr>
<td>Partial Evaluation of Model Transformations</td>
<td>Ali Razavi and Kostas Kontogiannis</td>
<td>University of Waterloo, Canada; National Technical University of Athens, Greece</td>
<td>562</td>
</tr>
<tr>
<td>Partial Models: Towards Modeling and Reasoning with Uncertainty</td>
<td>Michalis Famelis, Rick Salay, and Marsha Chechik</td>
<td>University of Toronto, Canada</td>
<td>573</td>
</tr>
<tr>
<td>Concurrency and Exceptions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Static Detection of Resource Contention Problems in Server-Side Scripts</td>
<td>Yunhui Zheng and Xiangyu Zhang</td>
<td>Purdue University, USA</td>
<td>584</td>
</tr>
<tr>
<td>Amplifying Tests to Validate Exception Handling Code</td>
<td>Pingyu Zhang and Sebastian Elbaum</td>
<td>University of Nebraska-Lincoln, USA</td>
<td>595</td>
</tr>
<tr>
<td>MagicFuzzer: Scalable Deadlock Detection for Large-Scale Applications</td>
<td>Yan Cai and W. K. Chan</td>
<td>City University of Hong Kong, China</td>
<td>606</td>
</tr>
<tr>
<td>Software Architecture</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does Organizing Security Patterns Focus Architectural Choices?</td>
<td>Koen Yskout, Riccardo Scandariato, and Wouter Joosen</td>
<td>KU Leuven, Belgium</td>
<td>617</td>
</tr>
<tr>
<td>Enhancing Architecture-Implementation Conformance with Change Management and Support for Behavioral Mapping</td>
<td>Yongjie Zheng and Richard N. Taylor</td>
<td>UC Irvine, USA</td>
<td>628</td>
</tr>
<tr>
<td>A Tactic-Centric Approach for Automating Traceability of Quality Concerns</td>
<td>Mehdi Mirakhorli, Yonghee Shin, Jane Cleland-Huang, and Murat Cinar</td>
<td>DePaul University, USA</td>
<td>639</td>
</tr>
<tr>
<td>Formal Verification</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Build Code Analysis with Symbolic Evaluation</td>
<td>Ahmed Tamrawi, Hoan Anh Nguyen, Hung Viet Nguyen, and Tien N. Nguyen</td>
<td>Iowa State University, USA</td>
<td>650</td>
</tr>
<tr>
<td>An Automated Approach to Generating Efficient Constraint Solvers</td>
<td>Dharini Balasubramaniam, Christopher Jefferson, Lars Kotthoff, Ian Miguel, and Peter Nightingale</td>
<td>University of St. Andrews, UK</td>
<td>661</td>
</tr>
<tr>
<td>Simulation-Based Abstractions for Software Product-Line Model Checking</td>
<td>Maxime Cordy, Andreas Classen, Gilles Perrouin, Pierre-Yves Schobbens, Patrick Heymans, and Axel Legay</td>
<td>University of Namur, Belgium; INRIA, France; LIFL-CNRS, France; IRISA, France; Aalborg University, Denmark; University of Liège, Belgium</td>
<td>672</td>
</tr>
<tr>
<td>Invariant Generation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Using Dynamic Analysis to Discover Polynomial and Array Invariants</td>
<td>ThanhVu Nguyen, Deepak Kapur, Westley Weimer, and Stephanie Forrest</td>
<td>University of New Mexico, USA; University of Virginia, USA</td>
<td>683</td>
</tr>
<tr>
<td>Metadata invariants: Checking and Inferring Metadata Coding Conventions</td>
<td>Myoungkyu Song and Eli Tilevich</td>
<td>Virginia Tech, USA</td>
<td>694</td>
</tr>
</tbody>
</table>
Generating Obstacle Conditions for Requirements Completeness
Dalal Alrajeh, Jeff Kramer, Axel van Lamsweerde, Alessandra Russo, and Sebastián Uchitel — Imperial College London, UK; Université Catholique de Louvain, Belgium

Regression Testing
make test-zesti: A Symbolic Execution Solution for Improving Regression Testing
Paul Dan Marinescu and Cristian Cadar — Imperial College London, UK

BALLERINA: Automatic Generation and Clustering of Efficient Random Unit Tests for Multithreaded Code
Adrian Nistor, Qingzhou Luo, Michael Pradel, Thomas R. Gross, and Darko Marinov — University of Illinois at Urbana-Champaign, USA; ETH Zurich, Switzerland

On-Demand Test Suite Reduction
Dan Hao, Lu Zhang, Xingxia Wu, Hong Mei, and Gregg Rothermel — Peking University, China; Key Laboratory of High Confidence Software Technologies, China; University of Nebraska, USA

Software Vulnerability
Automated Detection of Client-State Manipulation Vulnerabilities
Anders Møller and Mathias Schwarz — Aarhus University, Denmark

Understanding Integer Overflow in C/C++
Will Dietz, Peng Li, John Regehr, and Vikram Adve — University of Illinois at Urbana-Champaign, USA; University of Utah, USA

A Large Scale Exploratory Analysis of Software Vulnerability Life Cycles
Muhammad Shahzad, Muhammad Zubair Shafiq, and Alex X. Liu — Michigan State University, USA

API Learning
Synthesizing API Usage Examples
Raymond P. L. Buse and Westley Weimer — University of Virginia, USA

Semi-automatically Extracting FAQs to Improve Accessibility of Software Development Knowledge
Stefan Henß, Martin Monperrus, and Mira Mezini — TU Darmstadt, Germany; University of Lille, France; INRIA, France

Temporal Analysis of API Usage Concepts
Gias Uddin, Barthélémy Dagenais, and Martin P. Robillard — McGill University, Canada