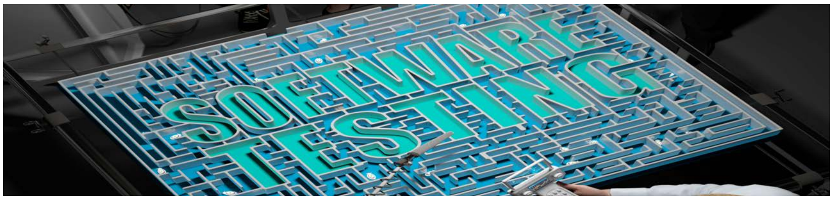
# Worlds Apart

## Industrial and Academic Focus Areas in Software Testing

**Vahid Garousi**, Hacettepe University

**Michael Felderer**, University of Innsbruck

*// A comparison of the titles of presentations in several industrial and academic conferences on software testing revealed different focus areas of industry and academia. This situation seems to be one reason for low industry–academia collaboration in software testing.//*



**INDUSTRY–ACADEMIA** collaboration (IAC) in software engineering (SE) has been an important topic since SE's early years (the 1960s). In an applied field such as SE, industrial impact and relevance[1] are crucial. For example, projects such as the ACM SIGSOFT Impact Project (www.sigsoft.org/impact.html) have measured and analyzed SE research's impact on practice.

However, many researchers and practitioners believe that the level of IAC in SE is low.[2] This is especially true for software testing, which is a hot topic in research and practice. The lack of mutual perception between industry and academia hurts both sides. Researchers have fewer insights on the problems that are important to practitioners, and practitioners fail to learn what researchers have already discovered that might be useful to them.[1]

Many calls for papers for academic SE conferences include wording such as, "This conference will bring together researchers and practitioners working in [insert any SE subarea here]." However, honestly, many conferences fail to really achieve that. Certain conferences have had some success—for example, the industry tracks of the International Conference on Software Engineering (ICSE) and International Conference on Software Testing, Verification and Validation (ICST). But much more must be done to really "bring researchers and practitioners together."

Toward that end, we focus here on software testing as a representative area of SE. To determine how industry and academia approach software testing, we compared the titles of presentations from selected conferences in each of the two communities. The results shed light on one cause of low IAC in software testing and led to suggestions on how to improve this situation. (For a look at what other researchers have done in this area, see the sidebar "Related Work in the Analysis of Software-Testing Research and Practice.")

## Our Approach

Figure 1 depicts our analysis approach. We selected three industrial conferences and two academic conferences on the basis of their representativeness and popularity in the community.

The industrial conferences were the Google Test Automation Conference (GTAC) and two regional versions of the widely known Software Testing, Analysis, and Review Conferences (EuroSTAR and STAREast). GTAC is Google's flagship test conference and has been held annually since 2006. The STAR series of conferences have been held annually in the US (STAREast and STARWest), Canada (STARCanada), and Europe (EuroSTAR) since the 1980s and 1990s.

0740-7459/17/$33.00 © 2017 IEEE

The academic conferences were ICST and the International Symposium on Software Testing and Analysis (ISSTA). ICST is supported by IEEE and has been held since 2008. ISSTA is supported by ACM and has been held since 1975 (first as a workshop with a different name; see historywiki.acm.org/sigs/SIGSOFT-ISSTA).

We collected the titles of the conference presentations from 2013 and 2014 (excluding satellite events such as workshops). This produced 354 industrial presentations and 340 academic presentations. The dataset is available at goo.gl/zK6KYw.

Next, we performed word cloud visualization and quantitative readability analysis to see the big picture and then qualitative analysis of a subset of the titles.

## Different Communities, Different Focuses

We used Wordle (www.wordle.net) to generate word clouds showing the focus areas of industry and academia based on the presentation titles. Figure 2 depicts the word clouds and lists the 10 most common terms. The visualization and top terms indicate a slight mismatch in focus areas. The top three terms for industry were "automation," "mobile," and "agile"; for academia, they were "model," "combinatorial," and "automated."

At a first glance, both communities appeared to focus on test automation (automated testing). However, the presentation titles reveal that when practitioners talked about test automation, they mostly meant automating test execution. In contrast, academics focused mostly on automated approaches in areas other than text execution (for example, test-case design).

## RELATED WORK IN THE ANALYSIS OF SOFTWARE-TESTING RESEARCH AND PRACTICE

Antonia Bertolino pointed out the large gap between the state of the art in the software-testing literature and the state of the software-testing practice.[1] She argued the need for further industrial empirical research in software testing.

Árpád Beszédes and László Vidács analyzed the synergies of academic and industrial software-testing conferences—for example, the composition of those venues' program committees.[2] Beszédes and Vidács also classified and compared the conferences on the basis of where they placed on the scale of purely academic, to joint works, to purely industrial.

Our research (see the main article) aimed to complement those papers. To analyze the focus areas of academics and practitioners regarding software testing, we looked at the titles of presentations at their respective conferences.

### References

1. A. Bertolino, "The (Im)maturity Level of Software Testing," *SIGSOFT Software Eng. Notes*, vol. 29, no. 5, 2004, pp. 1–4.
2. Á. Beszédes and L. Vidács, "Academic and Industrial Software Testing Conferences: Survey and Synergies," *Proc. 9th Int'l Conf. Software Testing, Verification and Validation Workshops* (ICSTW 16), 2016, pp. 240–249.

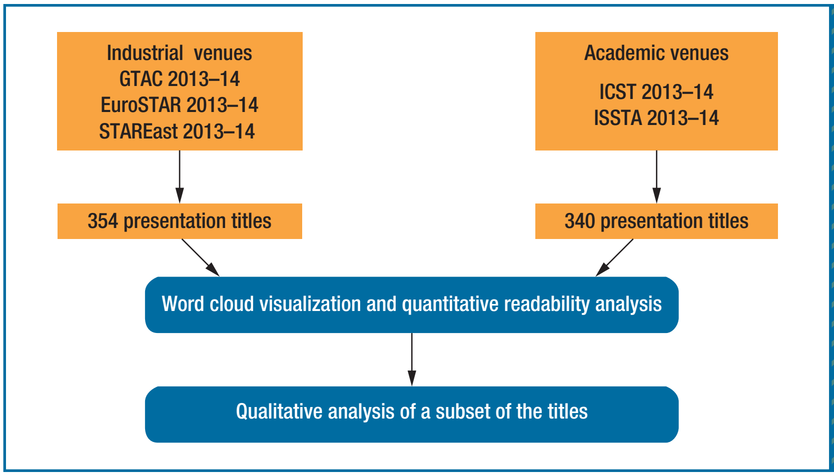
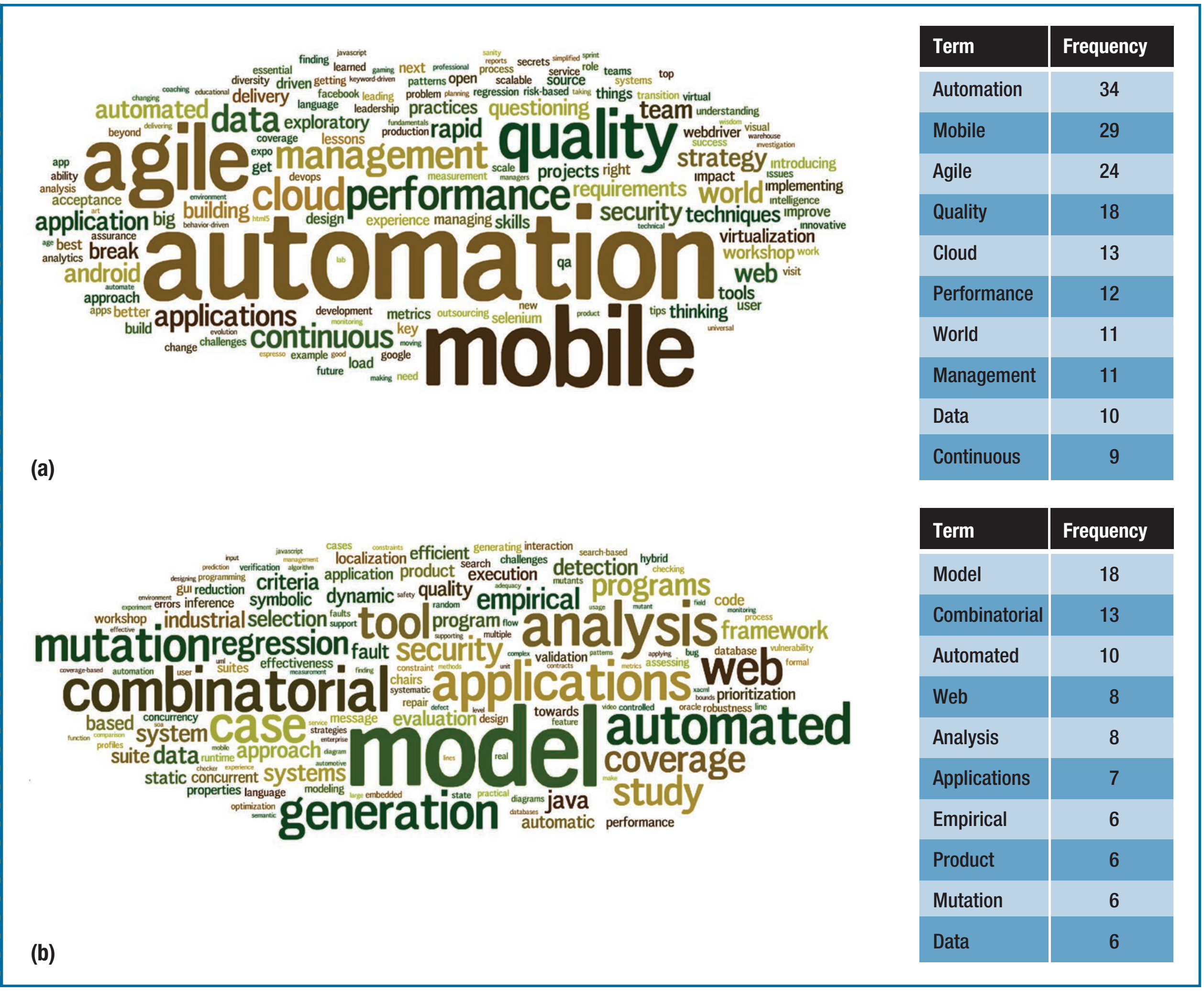


**FIGURE 1.** Our analysis approach. We chose the industrial and academic conferences on the basis of their representativeness and popularity. GTAC = Google Test Automation Conf.; STAR = Software Testing, Analysis, and Review Conf.; ICST = Int'l Conf. Software Testing, Verification and Validation; ISSTA = Int'l Symp. Software Testing and Analysis.



| Term | Frequency |
|---|---|
| Automation | 34 |
| Mobile | 29 |
| Agile | 24 |
| Quality | 18 |
| Cloud | 13 |
| Performance | 12 |
| World | 11 |
| Management | 11 |
| Data | 10 |
| Continuous | 9 |

| Term | Frequency |
|---|---|
| Model | 18 |
| Combinatorial | 13 |
| Automated | 10 |
| Web | 8 |
| Analysis | 8 |
| Applications | 7 |
| Empirical | 6 |
| Product | 6 |
| Mutation | 6 |
| Data | 6 |

**FIGURE 2.** Focus areas and the most common terms for the studied (a) industrial and (b) research conferences. Although both communities appeared to focus on test automation, they looked at it from different angles.

The example titles in Table 1 illustrate the disparate focuses on automated testing. The three industry presentation titles focus on automated test execution ("implementing automation," "virtualization"), its management ("outsourcing"), and its application in specific domains ("mobile devices," "financial institutions"). The three academic presentation titles focus on automated test generation ("automated search for probabilistic test profiles"), test evaluation ("automated generation of oracles"), and debugging ("automated program repair"), and those presentations took optimization techniques (search-based approaches) into account.

These results seem to indicate that the topics that interested the practitioners didn't interest the researchers, and vice versa.

Regarding the 10 most common terms, participants at the industrial conferences frequently discussed testing's relationship to software quality. Management issues and testing in the context of agile development were also hot topics. In addition, mobile testing, cloud testing, and performance testing were popular issues. Finally, testing in the context of continuous integration and delivery was frequently discussed.

Participants at the academic conferences were excited by challenging theoretical issues—for example, combinatorial testing and

TABLE 1

Example presentation titles from each type of conference, illustrating the different focuses on automated testing.

| Conference type | Presentation title |
|---|---|
| Industry | "The Challenges of Big Testing: Automation, Virtualization, Outsourcing, and More" |
| | "The Importance of Automated Testing on Real and Virtual Mobile Devices" |
| | "Designing and Implementing Automation at a Large Financial Institution" |
| Academia | "Using Automated Program Repair for Evaluating the Effectiveness of Fault Localization Techniques" |
| | "Automated Generation of Oracles for Testing User-Interaction Features of Mobile Apps" |
| | "Adding Contextual Guidance to the Automated Search for Probabilistic Test Profiles" |

search-based test-case design. On the other hand, participants at the industrial conferences usually discussed ways to improve testing effectiveness and efficiency (for example, using better test automation), without using "fancy" methods or techniques that they mostly found too complicated and hard to implement and deploy.[2]

Model-based testing was popular in academia. It did seem to have limited use in some industry sectors—for example, for testing automotive and safety-critical software. However, it had no widespread penetration in all industry sectors (for example, in testing mobile or web applications). Also, mutation testing was widely discussed in research but had low industrial penetration, according to the data based on the presentation titles.

We also discovered a difference in writing formality. Many practitioners believe that academic papers are too formal and hard to understand.[1,3] In contrast, practitioners usually strive to write things in the simplest form possible. Although this phenomenon's root causes are beyond this article's scope, we conducted an automated analysis on the presentation titles' readability, using a free online service (www.readability-score.com). (Because we didn't have transcripts of all the industry presentations, we couldn't analyze the presentations' full texts.)

Figure 3 shows the results for various well-known readability metrics. The results confirmed our expectation that the academic presentation titles were indeed harder to read than the industrial ones.

So, as you can see, industry and academia appear to live in two different worlds.

## Qualitative Analysis of a Subset of Titles

We manually reviewed a subset of the presentation titles to get a sense of the topics each group was interested in. We sorted the titles by their length and then analyzed the 10 longest and five shortest titles from each of the two groups (see Tables 2 and 3).

Most of the industrial presentations provided minitraining on specific topics (for example, presentations 4 and 6 in Table 2). Many presentations covered best practices and lessons learned (for example, presentations 2 and 3). Some presentations were about team building and human factors in this context (for example, presentations 7, 8, and 353). We also noticed presentations on the relationship between testing and process issues (for example, presentations 3, 7, 8, and 9). In addition, presentations on testing in current domains of interest such as mobile computing or robotics were common (for example, presentations 4, 350, and 352). Finally, the industrial presentations often used humorous titles, probably to attract an audience[4] and set an informal tone (for example, presentations 1 and 4).

The academic presentations focused more on theory (for example, presentations 6 and 337 in Table 3). There were many systematic empirical studies (for example, presentations 1, 5, 7, and 10). Some of these actually presented an evaluation in an industrial context (for example, presentation 5). Search-based testing was popular (for example, presentations 5 and 9), as was model-based test generation (for example, presentations 4 and 7). Presentations also covered security testing (for example, presentations 1, 3, and 4) and dynamic analysis (for example, presentations 2 and 336).

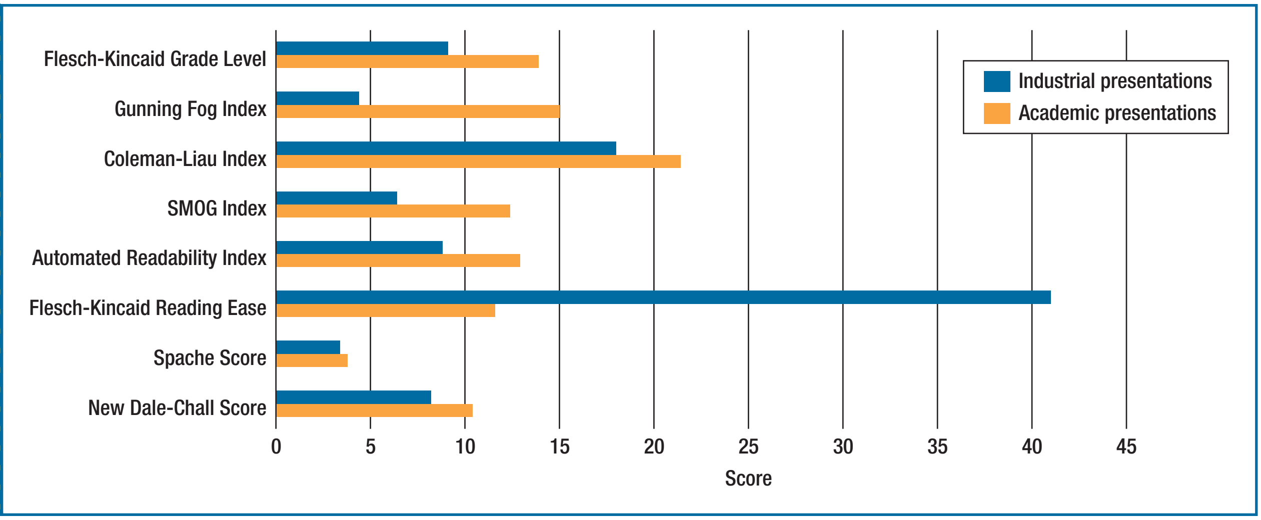


**FIGURE 3.** The presentation titles' scores for various readability metrics. The higher the score, the more difficult the titles were to read (except for Flesch-Kincaid Reading Ease, which is a reverse metric). The results confirmed our expectation that the academic conferences' presentation titles were harder to read than the industrial conferences' presentation titles.

**TABLE 2**

### The 10 longest and five shortest presentation titles from the industrial conferences.

| Rank | Title |
|---|---|
| 1 | "Free Tests Are Better Than Free Bananas: Using Data Mining and Machine Learning to Automate Real-Time Production Monitoring" |
| 2 | "Mobile Quality Assurance: What Functional and Non-functional Testers Need to Know about Advanced Best Practices" |
| 3 | "From Request to Delivery—Using Agile Methods to Trace Customer Requirements and Improve Quality" |
| 4 | "Espresso, Spoon, Wiremock, Oh My! (Or How I Learned to Stop Worrying and Love Android Testing)" |
| 5 | "Next Gen Automation: Abstracted Language and Tool Agnostic Approaches to Reduce Quality Costs" |
| 6 | "AddressSanitizer, ThreadSanitizer and MemorySanitizer: Dynamic Testing Tools for C++" |
| 7 | "Questioning Auditors Questioning Testing, or How to Win Friends and Influence Auditors" |
| 8 | "How We Transformed the Traditional Software QA by Getting Rid of the Central QA Group" |
| 9 | "The Art of Testing Transformation: Blending Technology with Cutting-Edge Processes" |
| 10 | "The Challenges of BIG Testing: Automation, Virtualization, Outsourcing, and More" |
| … | — |
| 350 | "Robotic Testing" |
| 351 | "Visual Testing" |
| 352 | "Mobile Testing" |
| 353 | "Team Building" |
| 354 | "Testing Me" |

TABLE 3

The 10 longest and five shortest presentation titles from the academic conferences.

| Rank | Title |
|---|---|
| 1 | "Empirical Investigation of the Web Browser Attack Surface under Cross-Site Scripting: An Urgent Need for Systematic Security Regression Testing" |
| 2 | "Make It Work, Make It Right, Make It Fast: Building a Platform-Neutral Whole-System Dynamic Binary Analysis Platform" |
| 3 | "Test Generation and Evaluation from High-Level Properties for Common Criteria Evaluations—the TASCCC Testing Tool" |
| 4 | "Generic Approach for Security Error Detection Based on Learned System Behavior Models for Automated Security Tests" |
| 5 | "A Search-Based Approach for Cost-Effective Software Test Automation: Decision Support and an Industrial Case Study" |
| 6 | "Test Generation for Robotized Paint Systems Using Constraint Programming in a Continuous Integration Environment" |
| 7 | "Assessing Quality and Effort of Applying Aspect State Machines for Robustness Testing: A Controlled Experiment" |
| 8 | "On an Embedded Software Design Architecture for Improving the Testability of In-Vehicle Multimedia Software" |
| 9 | "Search-Based Testing of Relational Schema Integrity Constraints across Multiple Database Management Systems" |
| 10 | "Threats to the Validity and Value of Empirical Assessments of the Accuracy of Coverage-Based Fault Locators" |
| . . . | — |
| 336 | "Collecting a Heap of Shapes" |
| 337 | "Mutant Subsumption Graphs" |
| 338 | "Reconstructing Core Dumps" |
| 339 | "Crowdsourcing GUI Tests" |
| 340 | "Declarative Mocking" |

## What We Can Do to Improve Things

Our analysis in this article and our experience in collaboration with industrial test engineers[5,6] show that the different focus areas are a key barrier to wider IAC in software testing and other areas of SE. If practitioners and researchers are interested in different topics, they'll be less likely to collaborate. (For a look at other reasons why IAC in SE is low, see the "Other Reasons for Low Industry–Academia Collaboration" sidebar.)

Many generic best practices to improve IAC in SE have been reported (for example, in our systematic review[2]) and should be carefully studied and implemented. For instance, researchers should focus on industry problems rather than abstract or artificial challenges, and practitioners should value the research results and be open to IAC.

This can be supported by empirical evaluations in industrial contexts—for example, using surveys or case studies—thus providing a link between industrial and academic topics in software testing. Such research studies integrate the industrial viewpoint. In addition, these studies can be presented as best practices and lessons learned at industrial conferences, which are especially welcome at these venues. In return, practitioners should raise sophisticated industrial challenges in which solutions can be expected from discussions with researchers or from special tracks at research conferences or workshops.

Researchers should employ the principles of *action research*,[7] especially when collaborating with industry, to ensure that the research problems are based on real industry needs. We recently proposed a grounded-theory-based approach for doing so in a paper[5] that was based on our experience in more than 15 IAC projects in testing.[6] Vahid Garousi and a major Turkish defense software company applied the approach

## OTHER REASONS FOR LOW INDUSTRY–ACADEMIA COLLABORATION

Researchers and practitioners have discussed reasons for the lack of motivation for industry–academia collaboration (IAC) in software engineering (SE),[1] such as each side having different objectives, industrial problems lacking scientific novelty or challenges, and the low applicability and scalability of solutions developed in academia.[2] Also, a 2015 survey of about 3,000 Microsoft employees showed that many practitioners found very few of the top-cited SE research papers relevant or useful.[3] There are even indications that the gap between industry and academia is bigger in software testing than in other SE areas (for more on this, see the "Related Work" sidebar).

Furthermore, most research papers lack cost–benefit analysis—that is, how much effort or time must be spent to adopt or implement a testing technique and how much real cost savings will result. Practitioners consider this lack a major limitation. (However, we've had initial success in conducting cost–benefit analysis of developing and applying new testing techniques in industrial settings.[4–6])

### References

1. V. Garousi, K. Petersen, and B. Özkan, "Challenges and Best Practices in Industry–Academia Collaborations in Software Engineering: A Systematic Literature Review," *Information and Software Technology*, Nov. 2016, pp. 106–127.
2. L. Briand, "Embracing the Engineering Side of Software Engineering," *IEEE Software*, vol. 29, no. 4, 2012, pp. 96, 93–95.
3. D. Lo, N. Nagappan, and T. Zimmermann, "How Practitioners Perceive the Relevance of Software Engineering Research," *Proc. 10th Joint Meeting Foundations of Software Eng.* (ESEC/FSE 15), 2015, pp. 415–425.
4. Y. Amannejad et al., "A Search-Based Approach for Cost-Effective Software Test Automation Decision Support and an Industrial Case Study," *Proc. IEEE 7th Int'l Conf. Software Testing, Verification and Validation Workshops*, 2014, pp. 302–311.
5. S.A. Jolly, V. Garousi, and M.M. Eskandar, "Automated Unit Testing of a SCADA Control Software: An Industrial Case Study Based on Action Research," *Proc. IEEE 5th Int'l Conf. Software Testing, Verification and Validation* (ICST 12), 2012, pp. 400–409.
6. S. Mohacsi, M. Felderer, and A. Beer, "Estimating the Cost and Benefit of Model-Based Testing: A Decision Support Procedure for the Application of Model-Based Testing in Industry," *Proc. 41st Euromicro Conf. Software Eng. and Advanced Applications*, 2015, pp. 382–389.

in ongoing IAC to derive the topics for several mutually attractive joint projects:

- providing more test automation for several test groups;
- assessing and improving an in-house test automation framework for a specific test group;
- establishing a systematic, effective, and efficient measurement program based on GQM (goal, question, metric) for the testing department;
- assessing and improving test process maturity using TMMI (Testing Maturity Model Integration) and TPI Next (Test Process Improvement Next); and
- enabling bidirectional knowledge transfer between the company under study and international conferences, workshops, and organizations in the aviation industry.

Research-intensive SE conferences must try to become more interesting from the industry's perspective and attract more practitioners. Some research conferences have made promising developments in this direction. Examples include ICSE's Software Engineering in Practice track, ICST's industry track, and the Workshop on Testing: Academia–Industry Collaboration, Practice and Research Techniques (TAIC PART).

Regarding industrial testing conferences, a good example of an attempt to increase IAC is the annual Software Quality Days in Austria, which hosts a scientific track with Springer proceedings. This track even has a best industrial-experience paper award, and many of the papers have a major case-study component.[8,9]

Researchers and practitioners must work to increase the level of joint collaboration and to ensure win–win situations for both sides. Researchers should also be aware of their industrial partners' challenges and choose problems and topics that are novel, feasible, industrially relevant, and potentially impactful.[10] Through studies such as this one, we're continuing our effort to bring practitioners and researchers in SE in general, and software testing in particular, closer to each other so that they can benefit each other much more than they do today.

Although the conferences we selected are among the most popular industrial and academic conferences, we don't claim that our results and observations are fully generalizable. Analyzing data from other conferences in the two categories might yield slightly different results.

## ABOUT THE AUTHORS

**VAHID GAROUSI** is an associate professor of software engineering at Hacettepe University and the managing director of Maral Software Consulting Corporation. His research interests include software testing, empirical software engineering, and improving industry–academia collaboration in software engineering. Garousi received a PhD in software engineering from Carleton University. He was an IEEE Computer Society Distinguished Visitor from 2012 to 2015. Contact him at vahid.garousi@hacettepe.edu.tr.

**MICHAEL FELDERER** is a professor at the University of Innsbruck's Institute of Computer Science and the managing director of QE LaB Business Services. His research interests include software and security testing, software processes, requirements engineering, empirical software engineering, and improving industry–academia collaboration. He also transfers his research results into practice as a consultant and speaker at industrial conferences. Felderer received a habilitation and PhD in computer science from the University of Innsbruck. Contact him at michael.felderer@uibk.ac.at.

## References

1. D. Lo, N. Nagappan, and T. Zimmermann, "How Practitioners Perceive the Relevance of Software Engineering Research," *Proc. 10th Joint Meeting Foundations of Software Eng.* (ESEC/FSE 15), 2015, pp. 415–425.
2. V. Garousi, K. Petersen, and B. Özkan, "Challenges and Best Practices in Industry–Academia Collaborations in Software Engineering: A Systematic Literature Review," *Information and Software Technology*, Nov. 2016, pp. 106–127.
3. "Why Are Academic Articles/Journals/Papers So Hard to Understand?," *English Language & Usage*, Stack Exchange, 2017; english.stackexchange.com/questions/132564/why-are-academic-articles-journals-papers-so-hard-to-understand.
4. "Titles That Talk: How to Create a Title for Your Article or Manuscript," Freelance Writing, 2017; www.freelancewriting.com/creative-writing/titles-that-talk.
5. V. Garousi and K. Herkiloğlu, "Selecting the Right Topics for Industry–Academia Collaborations in Software Testing: An Experience Report," *Proc. 2016 IEEE Int'l Conf. Software Testing, Verification, and Validation*, 2016, pp. 213–222.
6. V. Garousi, M.M. Eskandar, and K. Herkiloğlu, "Industry–Academia Collaborations in Software Testing: Experience and Success Stories from Canada and Turkey," *Software Quality J.*, 2016, pp. 1–53.
7. P.S.M. dos Santos and G.H. Travassos, "Action Research Use in Software Engineering: An Initial Survey," *Proc. 3rd Int'l Symp. Empirical Software Eng. and Measurement* (ESEM 09), 2009, pp. 414–417.
8. A.N. Ghazi, K. Petersen, and J. Börstler, "Heterogeneous Systems Testing Techniques: An Exploratory Survey," *Proc. 2015 Int'l Conf. Software Quality Days*, 2015, pp. 67–85.
9. M. Felderer and A. Beer, "Using Defect Taxonomies to Improve the Maturity of the System Test Process: Results from an Industrial Case Study," *Software Quality: Increasing Value in Software and Systems Development*, Springer, 2013, pp. 125–146.
10. A. Begel and T. Zimmermann, "Analyze This! 145 Questions for Data Scientists in Software Engineering," *Proc. 36th Int'l Conf. Software Eng.* (ICSE 14), 2014, pp. 12–23.