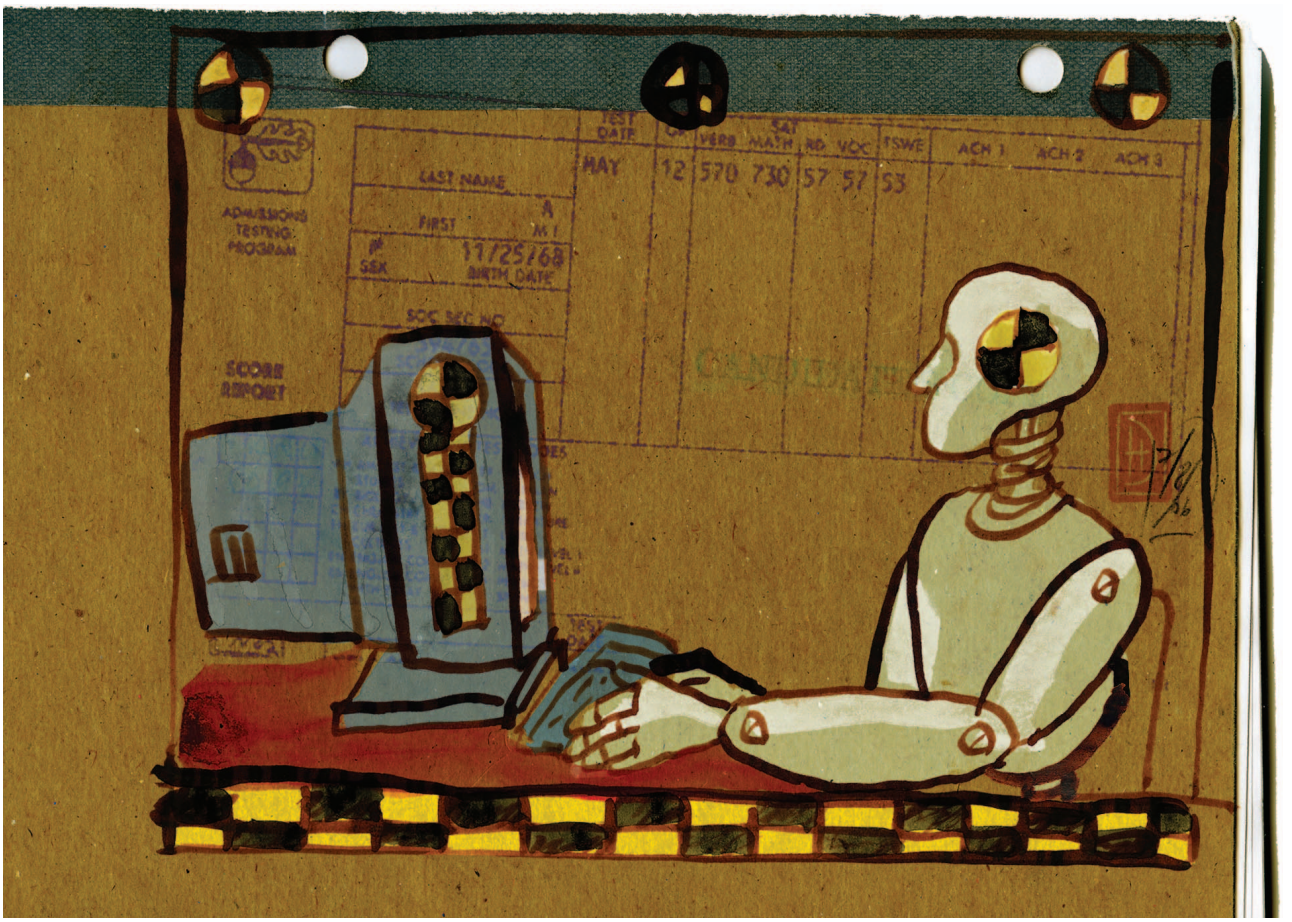
Software Testing Practices in Industry

Natalia Juristo and Ana M. Moreno, *Universidad Politécnica de Madrid*
Wolfgang Strigel, *QA Labs*

Constructing quality products continues to be one of software development's greatest challenges. Testing, one of the most crucial tasks along the software development life cycle, can easily exceed half of a project's total effort. A successful testing approach can



0740-7459/06/$20.00 © 2006 IEEE

Many software engineers see testing as a junior or entry position and use it merely as a springboard into development jobs.

save significant effort and increase product quality, thereby increasing customer satisfaction and lowering maintenance costs.

Despite these obvious benefits, the state of software testing practice isn't as advanced as software development techniques overall. In fact, testing practices in industry generally aren't very sophisticated or effective. This might be due partly to the perceived higher satisfaction from developing something new as opposed to testing something that already exists.

Also, many software engineers consider testers second-class citizens. They see testing as a junior or entry position and use it merely as a springboard into development jobs. However, this perception might be changing with the introduction of agile development techniques, in which testers become fully integrated members of the development team. The articles in this special issue demonstrate that testing is as much a professional discipline as development and can present interesting challenges and solutions.

Academia spends significant effort in researching new testing approaches. Promising approaches have started to find acceptance in industry, but the technology transfer between testing research and industry is still insufficient. Academics sometimes say that industry is immature and practitioners are clueless, whereas practitioners might argue that researchers squander their time developing cool but useless testing technologies. As often happens, the truth lies somewhere in between.

The challenges of unit testing

IEEE Software hasn't published a special issue on testing since 1991, so it's well worth revisiting this topic. We want to show how practitioners approach the challenges of testing—and particularly unit testing—and how such experiences can be useful to other organizations.

The IEEE defines *unit testing* as "the testing of individual software or hardware units or groups of related units." Other definitions refer to testing software components as opposed to a complete build or system test. The articles we chose all observe that developers rather than testers typically implement unit tests.

This issue

Software testing is a huge field; covering it fully in a special issue of four articles is impossible. We couldn't include many of the excellent papers that were submitted on the topic.

The articles we chose address a cross section of industrial testing techniques. The first, "A Survey of Unit Testing Practices," presents the results of a survey conducted in 19 Swedish companies to explore their unit-testing process. Per Runeson investigates what these companies consider a unit test, how these tests are performed, who executes these tests, when they are complete, and why companies perform such testing. The article identifies commonly observed problems in unit testing such as a lack of suitable documentation, lack of metrics, and the absence of clear criteria to indicate when you can stop testing. You might find this benchmark of interest for your own unit-testing process.

"Agile Software Testing in a Large-Scale Project" describes a radical departure from traditional testing approaches in the context of an agile development process. David Talby and his colleagues from the Israel Institute of Technology and Israeli Air Force describe their experience in applying agile testing in a large-scale, enterprise-critical software project. Their quantitative and qualitative data show how agile testing succeeded. Their field-tested tactics achieved promising results.

The last two articles deal with automating software testing. "Unit Tests Reloaded: Parameterized Unit Testing with Symbolic Execution" deals with how to use symbolic execution to increase code coverage by generating new unit tests and finding relevant variations. Nikolai Tillmann and Wolfram Schulte discuss how Microsoft adapted such techniques in developing commercial testing tools. The final article, "Industrial Deployment of the New TTCN-3 Testing Technology," analyzes the effective adoption of a relatively new scripting language in a large organization. Thomas Deiß and his colleagues discuss the various technical and management issues you must address when you introduce a new technology in an organization. They also touch on the development of new competencies and knowledge transfer and the other considerable benefits from adopting this new technology.

Finally, our roundtable discussion with several prominent researchers and industry consultants addresses the gap between testing research and industry practices. In particular, we asked them, Is the practice of software testing in industry effective? Not unexpectedly, this question raised some conflicting opinions. However, the experts all agreed that testing does not re-

ceive the same attention as development and that current industry testing practices leave a lot to be desired.

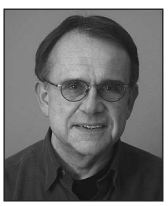
We hope that the industry experiences described here will help improve testing in other organizations. As long as industry does not increase its emphasis on testing practices, research into testing methods will remain limited. Similarly, the current underrepresentation of testing in software engineering curricula will continue.

On the other hand, we believe that the importance placed on testing will increase as software's pervasiveness in everyday life increases. Our dependence on software, from driving cars to shopping on the Internet, will decrease users' tolerance of defective software. Although testing isn't the only software engineering practice to ensure quality software, it will remain an essential component of the development life cycle.

About the Authors

Natalia Juristo is a professor of computer science at Universidad Politécnica de Madrid, where she coordinates the Software Engineering Group and directs master's courses in software engineering and knowledge engineering. She is a senior member of the IEEE Computer Society and member of the ACM, AAAS, and NYAS. Contact her at Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo s/n, E-28660 Boadilla del Monte, Madrid, Spain; natalia@fi.upm.es.

Ana M. Moreno is an assistant professor of computer science at Universidad Politécnica de Madrid. Contact her at Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo s/n, E-28660 Boadilla del Monte, Madrid, Spain; ammoreno@fi.upm.es.



Wolfgang Strigel is president of QA Labs. His special area of interest is software engineering process and, more recently, software testing in the context of pragmatic business needs. Contact him at QA Labs, #470-1122 Mainland St., Vancouver, BC V6B 5L1, Canada; strigel@qalabs.com.