have confidence in an implementation’s correctness. Symbolic execution techniques can help increase code coverage by finding relevant variations of existing unit tests. These techniques can also automatically generate unit tests from an implementation when no prior unit tests exist. Commercial testing tools are already adopting these techniques.

### A Survey of Unit Testing Practices
by Per Runeson, pp. 22–29. Most companies and testing books use the term unit testing, but its semantics vary widely in different organizations. Lund University researchers launched a survey in 19 companies to explore variations in the term. The results indicate a common understanding of the scope of unit testing, although its aim and practices vary across companies. Practitioners can use the survey instruments as a baseline for measuring their unit testing practices and to start improvement initiatives.

### Agile Software Testing in a Large-Scale Project
by David Talby, Orit Hazzan, Yael Dubinsky, and Arie Keren, pp. 30–37. Agile software development implies major changes in the way you test software, from strategic project planning to day-to-day activities. While previous publications have focused on programmer-oriented practices, such as test-first programming, they’ve neglected many issues related to large-scale projects and to an organization’s initial transition to agile methods. This article attempts to close this gap, reporting on the Israeli Air Force’s Extreme Programming experiences in a large-scale, enterprise-critical software project.

### Unit Tests Reloaded: Parameterized Unit Testing with Symbolic Execution
by Nikolai Tillmann and Wolfram Schulte, pp. 38–47. Unit tests are popular. However, it’s an art to write them in a way that specifies a program’s behavior well, and it’s laborious to write enough of them to have confidence in an implementation’s correctness. Symbolic execution techniques can help increase code coverage by finding relevant variations of existing unit tests. These techniques can also automatically generate unit tests from an implementation when no prior unit tests exist. Commercial testing tools are already adopting these techniques.

### Industrial Deployment of the TTCN-3 Testing Technology
by Thomas Deiß, Andreas J. Nyberg, Stephan Schulz, Risto Teittinen, and Colin Willcock, pp. 48–54. Increasing system complexity, reduced time-to-market, and stringent quality requirements pose challenges to software testing that are particularly prevalent in mobile telecommunications systems development. A standardized testing language like Testing and Test Control Notation 3 (TTCN-3) offers a possible solution to these challenges. However, introducing a new technology in a large organization poses its own complexities. This case study describes how Nokia introduced TTCN-3 technology throughout the company in a three-phase approach over five years.

### How Are Java Software Developers Using the Eclipse IDE?
by Gail C. Murphy, Mik Kersten, and Leah Findlater, pp. 76–83. Eclipse is a leading development environment that provides a rich set of features supporting Java development. However, little data is available about its usage: which features and plug-ins are developers using? Usage data from 41 developers using Java and Eclipse to support their daily work reveals that developers are using some advanced features, such as refactoring and third-party tools. However, they rarely use other features, such as bookmarking places in the code. Open source projects, such as Eclipse, should involve gathering and analyzing more usage data to ensure the tools evolve to meet the needs of their user communities.

### Project Visualization for Software
by Kai T. Hansen, pp. 84–92. Project team leaders can use software project “blueprints”—in the form of graphical expressions such as code- and function-oriented visualizations—to update stakeholders on the project’s progress through the development cycle. These graphical visualizations can represent artifacts produced in a software project. The author gives both theoretical and real-world examples showing how the visualizations map to architectural views and discusses how these visualizations could be linked to standard project management tools.