

# article summaries

JANUARY/FEBRUARY 2004, VOLUME TWENTY-ONE, NUMBER ONE

## DEVELOPING WITH OPEN SOURCE SOFTWARE

### The Many Meanings of Open Source

by Cristina Gacek and Budi Arief, pp. 34–40. Many software development methodologies are called “open source.” However simply stating that a project is open source doesn’t precisely describe the approach used to support the project. A multidisciplinary viewpoint can help determine those characteristics that are common to open source projects and those that vary among projects. These characteristics form the basis for a taxonomy of open source projects that’s useful for analyzing and setting up projects. They also provide a starting point for understanding what “open source” means.

### Mission-Critical Development with Open Source Software: Lessons Learned

by Jeffrey S. Norris, pp. 42–49. Using open source software components in a mission-critical project not only can keep the project within budget but can also result in a more robust and flexible tool. When considering an open source component, prospective users should evaluate the project for several characteristics: maturity, longevity, and flexibility. For greatest benefit, the users should also build and maintain a strong working relationship with the component’s developers.

### Developing an Information Systems Infrastructure with Open Source Software

by Brian Fitzgerald and Tony Kenny, pp. 50–55. Up to now, most open source software deployments have been in invisible infrastructure applications running on back-office servers (GNU/Linux, Apache, and so on). Beaumont Hospital in Ireland recently started developing its overall information

systems infrastructure by deploying more visible desktop and front-office open source applications in addition to GNU/Linux and Apache. In a two-phase open source implementation, Beaumont will save over €20 million over five years.

### From Proprietary to Open-Source Tools in Information Systems Development

by Nicolás Serrano, Sonia Calzada, Jose Mari Sarriegui, and Ismael Ciordia, pp. 56–58. Over the past decade, the University of Navarra’s Academic Management System evolved from a mainframe environment to a Windows-based client-server architecture and then to a Web environment. Aiming for independence from vendors, these developers adopted open-source solutions for their Web applications and were delighted with the results.

### Free and Open Source Development Practices in the Game Community

by Walt Scacchi, pp. 59–67. Findings from empirical studies of free and open source software systems in different communities show that some common processes and practices exist across the board. The studies focused on software development practices, social processes, technical system configurations, organizational contexts, and interrelationships that give rise to free and open source systems. These distinct communities, and the computer game community in particular, provide examples of common practices.

### New Tricks: How Open Source Changed the Way My Team Works

by Stephane Lussier, pp. 68–72. In 1998, a commercial software team at Macadamian Technologies contributed to Wine, an open source implementation of the Windows API on X-Windows and Unix. Expecting chaos in the organization and code, the team was surprised to find a structured

community with procedures all its own. After overcoming their prejudice, the professional developers learned that the open source community might have a few things to teach commercial software teams.

### Economics of Linux Adoption in Developing Countries

by Nir Kshetri, pp. 74–81. Software globalization affects build-versus-buy decisions at every level and phase of the software development process. One of the most interesting and controversial instances of software globalization is the expansion of Linux, an open source operating system, into software development efforts in developing countries. This article examines the positive and negative effects of adopting Linux and ways to achieve the greatest overall benefits.

### Using Open Source Software in Product Development: A Primer

by Michel Ruffin and Christof Ebert, pp. 82–86. When you integrate open source into your software project, you must understand the legal aspects of using it in commercial products and how to manage them to mitigate related risks. On the basis of practical experience, the authors answer the questions their coworkers have asked when confronted with the task of introducing open source to their teams.

## FEATURE

### Rapid Prototyping for a Virtual Medical Campus Interface

by Andreas Holzinger, pp. 92–99. The developers of Graz University’s Virtual Medical Campus, working under a strict timeline, used simple, rapid, cost-effective prototyping techniques to create a user interface and release a working system within six months. Involving users early in the interface design facilitated acceptance.