

SOFTWARE ENGINEERING IN THE INTERNET AGE

As an enabling technology, the Internet changes software engineering practice by giving “just in time” processes economic viability. These processes help companies meet the challenges of developing software under tightening market conditions.

FRANK MAURER

University of Calgary

GAIL KAISER

Columbia University

The Internet is one of the most influential factors in today's software development activities. Part of its influence is as a common target platform. Countless software engineering projects are now addressing issues such as access to legacy systems via the Web, development of non-Web client-server applications based on TCP/IP (or higher level protocols that are mapped down to it), and so on. The technology used or devised for these purposes defines a buzzword set for the Internet age: Java, HTTP, HTML, XML, CORBA, ASP, JDBC, scripting languages—to mention only a few.

But the Internet is also an enabling technology that allows companies to meet the challenges of developing and evolving software under tightening market conditions, where getting to market first can be more important than actual development costs. The Internet supports globally distributed product development so that work can proceed around the clock in different locations.

Furthermore, as distributed work becomes “the norm in most large, multinational companies,”¹ the Internet supports the formation of virtual enterprises for a specific project. Members of a virtual enterprise nevertheless face the challenge of integrating heterogeneous processes. This challenge imposes requirements on software process support tools to

- adapt to the virtual enterprise and its processes,
- proactively and automatically support process changes,
- support both synchronous and asynchronous activities and workflows,
- provide fast, transparent access to all information needed to carry out software development while respecting proprietary and security boundaries in organizational data and processes, and
- support the integration of different organizations at a technical level that avoids paper-based information exchange.

The Internet provides a nearly ubiquitous communication infrastructure that enables team members to connect to the development process with little

effort. It will shortly be accessible from every home as well as in the office and in between. Both telecommuters and mobile road warriors will be able to connect to anywhere in the world to fetch information and deliver work results.

Integrating process modeling and enactment tools with project management packages can give distributed project team members up-to-date information about the state of the work process as well as guidance about what to do next. Work processes can then be coordinated and issues tracked on the Web. Techniques from computer-supported collaborative work (CSCW) can be used to build virtual group spaces on the Web, providing meeting rooms as well as repositories of work products. Software artifacts can be passed on to co-workers around the globe with a few mouse clicks. Software development tools can be supplied as Web services. Collecting process and product metrics can be supported on the Net along with the interpretation of the gathered data. Experience bases can be provided on the Web to bring software process knowledge to team members facing new problems.

These opportunities mean that the Internet influences *how* software is developed—and that is what this special issue is all about: Software Engineering Over the Internet.

EFFECTS ON EVERYDAY PRACTICE

How is everyday software engineering practice different when the Internet is used? Software developers must still carry out the same activities as before: requirements analysis, design, programming, testing, configuration management, and so on.

Asking the same question of telephones and fax machines illuminates the answer. Can you imagine software development without these tools? No calls to a colleague to clarify a specific design decision? No fast answers about a subtle ambiguity in a client's requirement?

In the same sense, the Internet as an enabling technology changes software engineering processes. Most of these changes do not *require* Internet technology. Many of them could have been part of daily practice even in the old days of snail-mail and parcel services. But using the Internet as the base infrastructure gives these new "just-in-time" processes economic viability as a way of doing business.

Let's consider two examples of how Internet technology is changing software engineering practices.

One is beta testing. The Internet provides a cheap way for distributing thousands of copies of

new software to beta testers. Their feedback helps to resolve problems and improve the program. Microsoft, for example, is riding this wave extensively. And a nice touch: these beta testers provide their services free of charge. In principle, assuming the software itself was Internet-enabled, applications could even be instrumented to be self-testing, perhaps employing field data when security isn't a concern, and automatically transmit fault reports back to the development organization.

Another example is open-source projects, which offer an alternative to conventional software engineering processes for some kinds of software development efforts, but not all (though some open-source proponents may disagree with this caveat). These projects share source code freely among many "hobby" programmers and let the programmers—more or less without coordination—extend and improve it.

Eric Raymond² has analyzed how and why this bazaar-like development model works and how, for example, the evolution of Linux demonstrates the high-quality software results from such a joint effort. Netscape's decision to give away the source code for its Communicator tool was influenced by Raymond's paper. The free Apache Web Server is another example of an open-source, commercial strength project (for a summary of this project, see the Collaborative Work column from the July/August 1997 issue of *IEEE Internet Computing*³).

THE ARTICLES

The articles in this special issue include revised workshop manuscripts as well as independent submissions to the magazine. Four articles were selected from the peer-review process.

"Web-Based Issue Tracking for Large Software Projects," (pp. 25-33) by John R. Callahan, Reshma R. Khatsuriya, and Randy Hefner, discusses a Web-based, issue-management tool for NASA's Earth Observing System Data and Information System project. The tool tracks software problems over the Internet, formally documenting problems; describing the cause, impact, and suggested fixes; and employing issue-tracking information to determine project quality. An independent verification and validation contractor uses this tool to manage issue reports during development. The tool also provides data that is used for process improvement, progress evaluation, and decision making.

"Building a Software Design Laboratory on the Internet," (pp. 41-48) by Stefano Ceri, Piero Fraternali, Stefano Gevinti, and Stefano Paraboschi, shows how the Web can be used as a uniform and

ubiquitous interface to software engineering applications. It demonstrates how new software-design support tools can be made available on demand over the Web. Internet access to tools may improve the use of software engineering technology, especially for smaller scale enterprises. Instead of being forced to buy a general-purpose tool and maintain it on site, they could just buy a service to solve a specific problem when they need it. The authors show how their approach works by providing tools for analyzing, designing, and implementing database applications via the Web.

"A Decentralized Architecture for Software Process Modeling and Enactment," (pp. 53-62) by John C. Grundy, Mark D. Apperley, John G. Hosking, and Warwick B. Mugridge, presents a tool that supports process modeling and enactment synchronously as well as asynchronously. The authors discuss the shortcomings of centralized client-serv-

er approaches to supporting software process modeling and enactment—namely, robustness, performance, and security problems. They developed a decentralized architecture for process modeling and execution support that incorporates distributed work coordination and task automation. Their visual, multiview description of software development processes supports enactment awareness; and their workflow environment provides a robust, fast system for coordinating work processes utilizing basic Internet communication facilities.

"A Web-Based Tool for Data Analysis and Presentation," (pp. 63-69) by Roseanne Tesoriero and Marvin Zelkowitz, describes an environment that facilitates the understanding of distributed software metrics data. The authors argue that the ubiquity of the Web increases the likelihood of distribution of software processes among physically separated locations. Therefore, project metrics will also be stored in several locations and potentially differing formats, increasing the difficulty of gathering and interpreting data. Their WebME data visualization tool permits the display and analysis of development data in distributed, heterogeneous environments.

FURTHER READING

The WebDAV project is concerned with proposing extensions to HTTP that facilitate Web-based authoring and document management when multiple persons may edit the same document, which is common with design documents, source-code modules, and so on. It was described in a Collaborative Work column last year ("Distributed Authoring and Versioning," Vol. 1, No. 2, March/April 1997, pp. 76-77) and is updated in this issue ("WebDav," pp. 34-40).

Two recent workshops addressed issues related to Internet-based software development projects:

- "Software Engineering over the Internet" was part of the 20th International Conference on Software Engineering (ICSE 98) in Kyoto, Japan. Links to papers from the workshop are available at <http://sern.ucalgary.ca/~maurer/ICSE98WS/ICSE98WS.html>.
- "Coordinating Distributed Software Development Projects" was held in conjunction with the IEEE 7th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 98) in Stanford, California. Abstracts for the accepted papers are available at <http://www.gr.informatik.uni-kl.de/~holz/WETICE98>; proceedings are being published by IEEE Computer Society.

URLs for this Introduction

Apache Web Server • www.apache.org/

CSCW (one place to start a search is "Team IT - The Forum for CSCW") • www.csc.liv.ac.uk/~team-it/index.html

Linux • www.linux.org/

CHANGING THE WAY WE DEVELOP SOFTWARE

There are several commercial and research projects under way in addition to those reported in this issue (see the sidebar, "Further Reading"). They promise better methods, techniques, and tools to support software engineering over the Internet in the future. *IEEE Internet Computing* magazine will continue to publish research and reports on these projects in regular articles and columns, as they change the way software is developed, deployed, and maintained. Stay tuned. ■

REFERENCES

1. T. Athey, "Leadership Challenges for the Future," *IEEE Software*, Vol. 15, No. 3, May/June 1998, pp. 72-77.
2. E. Raymond, "The Cathedral and the Bazaar," paper first presented at Linux Kongress, 1997; available online at <http://www.earthspace.net/~esr/writings/cathedral-bazaar/>.
3. R.T. Fielding and G. Kaiser, "The Apache HTTP Server Project," *IEEE Internet Computing*, Vol. 1, No. 4, July/Aug. 1997, pp. 88-90.

Frank Maurer and Gail Kaiser are members of the editorial board for *IEEE Internet Computing*. Links to their home pages can be found at <http://computer.org/internet/>.