



A FutureU™ Industry Review

## **OPEN-SOURCE COLLABORATION SOFTWARE**

### **Is It Ready for Prime Time?**

Authors:

Claude Whitmyer, CLO and Chief Community Strategist

Gail Terry Grimes, CEO and Chief Creative Officer

Release: Summer, 2004

Copyright © 2004 by Claude Whitmyer and Gail Terry Grimes. All rights Reserved.

Published by FutureU™ ([www.futureu.com](http://www.futureu.com)) San Francisco, California.

# OPEN-SOURCE COLLABORATION SOFTWARE:

## Is It Ready for Prime Time?

### I. Overview

For more than ten years, FutureU has been advising organizations on the selection of software for virtual collaboration and distance learning. Questions increasingly arise about the long-term efficacy of the most popular collaborative tools. Concerns include:

- The cost of products and support
- Steep learning curves
- Complexity (Many more features than the average user will ever master)
- The difficulties of customization (The need to first learn a complex proprietary programming language)

The search for alternatives is ongoing. One promising direction is open-source software.

For a comprehensive definition of this phenomenon, go to

<http://www.opensource.org/docs/definition.php>.

To summarize, open-source software is written in such a way that the user may easily view the source code from which the product was created. There is an implied expectation that the user may modify and/or distribute the product and its source code at will. In fact, most proponents of open-source fervently believe that users should actively share such software and continuously strive to improve it. This is the opposite of the proprietary approach, in which the source code remains a well-guarded secret, and sharing is punishable by law. Software developers who have made a large

investment of time and money, and are hoping for a return on their dollar, tend to hold their source code particularly close to the chest.

FutureU Chief Learning Officer Claude Whitmyer recently undertook to identify and test the most promising open-source products available for supporting collaboration among team members who are physically distant from one another. His goals were to:

- Clarify whether and how open-source products might meet the need for virtual collaboration.
- Identify which open-source tools, if any, are the most efficient, cost effective and easiest to learn.
- Identify the current limitations of open-source products and gauge the likelihood and possible timing of significant improvements.
- Reduce the long-term software expenses of FutureU clients and colleagues.

This research was a collaborative undertaking with FutureU client colleagues at the NASA Astrobiology Institute (NAI), specifically the NAI Information Technology Working Group. Sincerest thanks go to the whole group and especially to Bertil Olsson, Frank Kachurak, John Armstrong, Uwe Rossbach, and Val Watson for their participation. Particular thanks go to Dr. Gotthard Saghi-Szabo of the Carnegie Institute of Washington, D.C., for his many contributions.

## **II. Method of Information Collection**

It seemed desirable from the outset to identify which features and activities participants in this investigation would be willing to explore. Several members of the NAI IT Working Group contributed useful information about their own experiences using open-source software for collaboration and document management; in particular, Dr. Bertil Olsen from the Marine Biological Laboratory at Woods Hole, Massachusetts, presented a demonstration of his use of Post NUKE. Mr. Whitmyer searched the Internet, visiting promising web sites, entering online demonstrations and examining lists of software features. Dr. Saghi-Szabo did the same and went a step further by attempting to test a few of the products that seemed most promising. Although these colleagues shared a particular curiosity about applications for scientific research, most of the software features that sparked their interest would be just as useful to any virtual community.

### **III. Discussion**

#### ***A. Features***

Previous research has identified the following as potentially valuable features in any virtual collaboration environment:

- Web-based emailing lists
- Video desktop computer tools
- Web-based photo directory
- Web-based information repository (to archive information)
- Visualization/imaging capabilities
- Room with videoconferencing screen and camera
- Wireless data sharing to/from remote sites
- Web-based document sharing capability
- Data sharing desktop computer tools (e.g., NetMeeting, WebEx, Centra, Placeware)
- Live chats (secure on-line spaces for real time discussion)
- Document sharing by posting the text to a Web site for download
- Asynchronous threaded discussions (secure on-line spaces for group discussion over time)
- High-resolution video (e.g. HDTV)
- Remote collaboration utilizing specific instruments (e.g. electron microscope)
- Information repository and data-sharing tool for distributed teams
- Audio-only desktop computer tools (Voice over Internet)
- Room with SmartBoard or other large white board

- Idea sharing via collaborative, modifiable, semi-graphical presentations and discussion space (e.g. concept mapping tools such as IBIS)
- Shared applications
- Virtual reality immersive environments (e.g., 3D surround environments)
- PowerPoint presentations
- PDF documents
- Text documents attached to email
- Documents via FedEx or mail
- E-mail correspondence
- Faxed documents
- Text documents by floppy disks, CDs, zips

### *Other Features: The "Short List"*

The following set of core features was identified as the essential starting point for any small collaborative team:

- Calendaring
- Tasks
- Memos
- Project Management
- Workflow Management
- Issue Tracking & Trouble Tickets
- Document Sharing/Version Control
- Notifications/Subscriptions
- Group Folders

- Chat Spaces with Breakout Rooms

## **B. *Software Categories***

No open-source package emerged as providing all the desirable core features in a single integrated offering. The feature sets that appeared most often can be roughly grouped into three categories:

### **1. Personal Information Management**, including:

- Email
- Calendaring and Discovery of Free Time
- Scheduling of Meetings
- Task Lists
- Address Books
- Instant Messaging

### **2. Content Management**, which generally included:

- Document Management (easy storage and retrieval)
- Weblogs (web-based journal or diary)
- Twikis (web-based jointly authored documents)
- Content management suites generally did not integrate collaboration tools.

### **3. E-Mail Lists** (e.g., MailMan and NewsGroups)

E-Mail lists tend to stand alone and, to the dismay of anyone looking for simple solutions, are generally not well integrated into either collaboration tools or content management tools.

### *C. Possible Contenders*

There are, nonetheless, three open-source suites that offer a broad set of features, with many of the features appearing on the above list of the most desirable features for collaboration. The contenders are:

- PHPNuke ([phpnuke.org](http://phpnuke.org))
- Zope ([zope.org](http://zope.org))
- TikiWiki ([tikiwiki.org](http://tikiwiki.org))

#### *PHPNuke ([phpnuke.org](http://phpnuke.org))*

PHP Nuke is primarily a content management system. The development community that supports it offers many collaboration modules, including a “news grabber” that allows you to scan other sites and add their news to your own site; groupware, including IMAP and POP3 email, Usenet newsgroups, contact management, scheduling, shared notes and bookmarks, to-do lists and meeting announcements; rosters; blogs; chat; auctions (eBay-like); calendars; event management; file management; link management; image galleries and ecommerce.

#### *Zope ([zope.org](http://zope.org))*

Zope is an open-source application server for building content management, intranets, portals, and custom applications. The Zope community is one of the largest open-source communities, with hundreds of companies and thousands of developers all over the world. Zope is written in Python, an object-oriented scripting language. Zope offers a large number of modules by function, including, e-commerce, database adapters, feedback solicitors, site-building templates, navigational aids, server utilities, user management, and visual tools.



Specific tools include wikis, blogs, mailing lists, newsletters, mail archive, address book, a full-text index product, database report generators, and more.

### *TikiWiki (tikiwiki.org)*

TikiWiki is primarily a content management system, but it also has a well-integrated groupware feature. TikiWiki has been used to create web sites, portals, and intranets. It incorporates many of the features available in most wiki systems, allowing a range of complexity from a simple newspaper-like wiki to a complex site suitable for use by a whole user community. Major features include: articles, forums, newsletters, blogs, file/image galleries, wikis, drawing, tracking, directories, polls/surveys/quizzes, FAQs, chat, banners, webmail, calendars, and classification of objects by category

## IV. Conclusions

**Current Benefits.** This investigation identified a number of potential benefits to using open-source software.

- The startup cost is negligible.
- Resources on the internet can provide a significant amount of information for a group wishing to implement an open-source solution.
- Open-source software can be robust, and sometimes is even more stable than its closed-source counterparts.
- Most open-source solutions use open standards, making them easily compatible with other applications that use the same standards.
- Open-source solutions offer flexible combinations of features, and can frequently be customized. If a community cannot find a version that meets its needs, a package with the desired feature set *can* be engineered.
- Open-source software developers tend to offer enthusiastic support, often at no charge.

**Current Drawbacks.** This investigation also uncovered several potential drawbacks to open-source software:

- Certain popular commercial packages, most famously Microsoft Office, are incompatible with open-source solutions.
- Support for open-source software can be difficult to interpret, intended, as it often is, for developers rather than end users. In fact, many open-source projects have no end-user documentation at all. Even when such documentation exists, it

may be outdated, lacking in detail, and decipherable only by the most tech-savvy developer.

- Open source is not plug-and-play. Loading, installing, and customizing the software can be a major hurdle. Modern hardware, technologies, and closed-source communication standards often are not supported in open-source packages.
- Many open-source projects are quite new. To ensure the continuity of a project based on open-source tools, the user group may need to hire exceptionally capable professional computer programmers, support staff and information systems staff.
- Untested waters are seldom smooth sailing when a bureaucracy is involved. Large institutional virtual communities should brace for a longer ramp-up time and extra layers of review and approval when seeking to adopt a full-function, secure, open-source environment.

**Assessment of Current Financial Picture.** Short-term, direct costs associated with open-source solutions are much lower than for a comparable commercial software package. In the long term, however, certain indirect costs associated with maintaining a large-scale, open-source package—such as the tangible cost of formal training, development of applications, and downtime—can easily exceed the cost of commercial software maintenance.

**Predictions.** This is a rapidly changing field. It is safe to say that not too much time will pass before open-source solutions become ideally suited for collaborative environments, for the following reasons:

An open-source license does not restrict the buyer from selling or giving away the software, either as a single software package or as a component of an aggregate software package that contains programs from multiple sources.

The distributed program must include source code and must allow distribution of source code along with the compiled program.

- Open-source software must not discriminate against any person or group of persons.
- Open-source licenses do not restrict anyone from making use of the program in a specific field.
- The license does not place restrictions on other software that is distributed along with the licensed software.
- Finally, no provision of the software may be predicated on any individual technology or style of interface.

**Conclusion.** In summary, it was determined that open-source software is not quite "ready for prime time." Generally, installation and maintenance remains excessively labor intensive. User support and documentation is uneven. As a result, relying on open-source software as a major collaboration platform would require a team of developers to test, tailor, implement and manage the installation. Only those organizations entirely unable to meet their needs with proprietary products, and with extensive financial and manpower resources to apply to the task, should consider such an installation at this time.

## V. Recommendations

There are many open-source products, ranging in scope and comprehensiveness, with a corresponding range of ease of installation, use, and administration. At the present time, however, because of programming and support issues, open-source software is not easily scalable. Creating full feature sets like those offered by commercial products would be labor intensive indeed. However, open-source software could be useful to small groups who have the technical expertise to tailor the software for their own use.

As listed above, at least three open-source products do merit further consideration by those seeking a solution today: PHPNuke, Zope and TikiWiki. Budget and priorities permitting, any one of these products may fit the needs of a small, technically savvy group. A close comparison of the three products is advisable, followed by piloting. The value of a pilot cannot be overestimated, not only to refine the tools themselves but to address the human element that comes into play whenever a group attempts to introduce change.