# Continuous Design of Free/Open Source Software

## Preliminary Workshop Report and Research Agenda

Les Gasser[1,2] and Walt Scacchi[1]
[1]Institute for Software Research
University of California Irvine
and
[2]Information Systems Research Laboratory
Graduate School of Library and Information Science
University of Illinois, Urbana-Champaign
(gasser@uiuc.edu, wscacchi@uci.edu)

15 October, 2003

## Summary

The PIs received an NSF research grant (IIS-#0350754) for a collaborative project and workshop focused on identifying research directions for Continuous (Re)Design of Free/Open Source Software. This report presents the preliminary results from a workshop held in two locations (UCI, 23 Sept 03 and UIUC, 8-9 Oct 03). The aim of this distributed workshop was to explore and organize the sense of the community in this research area, in relation to establishing a new "Science of Design." A total of 46 academic scientists and industrial researchers from across the U.S. came together in the workshop (18 at UCI, 28 at UIUC) to discuss, debate, and identify the critical issues and research directions. This preliminary report summarizes the findings of the workshop in five areas: motivations; areas/topics needing research; needed research collaborations and shared research infrastructure/test-beds; required investment; and other recommendations for action. As critical research in this area proceeds, it will have significant scientific, technological, educational, social, and economic benefits to the U.S.

## *Background*

Two related themes came together in the planning for this workshop: the ideas of "continuous design" and of so-called "open-source software" (OSS) development. There are many understandings of "open source software" (e.g., see http://www.opensource.org for a license-oriented perspective). Some of the key defining characteristics of OSS design/development processes and the software that they create, as these appear in general practice, include:

- Freely available, openly shared, clearly licensed/attributed source (and binary) code.
- Geographically-distributed, temporally asynchronous development processes.
- Community-based development and community-oriented approaches to project organization.
- Commitment to open standards for design representation and interaction.
- Design practices enabled by open, standards-based, Internet-enabled tools and infrastructure.

Many OSS projects and software artifacts are in regular widespread use supporting critical applications and continuously-operating infrastructure, including the Internet and Web themselves. OSS projects are having significant impacts in spheres of scientific, technical, artistic, and economic innovation and development, including astrophysics and deep space imaging; online computer games and the entertainment industry; internet infrastructure; business and e-commerce; and information technology research. Indeed, it is clear that the new NSF thrust in CyberInfrastructure will, in many ways, rest on an OSS foundation.

The communities of OSS users and developers are often interwoven. The deep engagement of users and developers, coupled with the openness of systems (in terms of both standards and access), means that system design and re-design activities are pursued continuously by community members. This happens through many concurrent channels, over the entire lifecycles of systems. It is often facilitated by communication and knowledge-sharing infrastructures such as persistent chat rooms, newsgroups, issue-reporting/tracking repositories, sharable design representations and many kinds of "software informalisms". This ongoing "continuous design" activity isn't entirely specific to OSS. However, it is prevalent in OSS communities, and is a novel and important emerging trend. Recognition of this (in several existing research projects) led to the dual focus of the workshop.

Finally, the core philosophies, actual practices, and practical applications of continuous design and OSS are having impacts far beyond those in computing and software development. They are in fact fostering novel approaches in many arenas central to the knowledge economy including artistic design, publishing, knowledge organization, and commerce.

## Preliminary Workshop Findings

Below, we sketch the workshop's central findings in five areas:

- Motivations for investing in continuous design/OSS research
- Key areas/topics needing immediate research investment
- Needs for research collaborations and shared research infrastructure
- Required investment in research
- Other recommended actions on the part of NSF and others.

A set of PowerPoint slides representing the findings in this report will be provided shortly, and a more detailed workshop report is under preparation.

## Motivations for Investing in CD/OSS Research

The workshop participants found seven main motivations for CD/OSS research, as follows:

### *Surprising impact, not well understood*

Software is central to the functioning of modern society. The OSS phenomenon is novel, widely-growing approach to developing both applications and infrastructure software that exhibits many counter-intuitive dimensions and is not well understood. For example, factors such as unpaid participation, open sharing of work products without financial gain, and the ability of highly-informal loosely-organized collectives to produce highly-reliable software, are phenomena of high potential scientific, educational and economic interest, for which we don't yet have adequate accounts.

### *Trusted and high-confidence systems*

Open artifacts and processes play a role in establishing and maintaining trustworthiness and confidence of complex and core-infrastructure systems, which is an existing national research thrust. The peer-review process for reducing the risk of research investment and improving confidence in scientific findings is a clear analog. Critical systems such as electronic voting systems, rights-management technologies, internet/e-commerce infrastructure, and scientific experimentation support have all gained direct improvements in trustworthiness and user-confidence through OSS methods, and research can increase the benefit.

***Competitive advantage***

Understanding, controlling, and improving the real effectiveness of OSS is a basic competitive issue for the national economy given emerging software investments and policies worldwide. Claims of OSS's ability to produce complex artifacts "better, faster, cheaper" are provocative but it is important to understand whether (and how) they can be realized. Moreover, OSS is an "innovation frontier" providing a very significant segment of scientific and technical growth. The OSS innovation engine operates under a novel economic calculus that may be exploitable in other settings if it is better understood.

***Fundamental to critical infrastructure development and participation.***

The OSS concept has been the basis for core elements of the critical infrastructures of the knowledge economy such as Internet and the Web (and their contents). One simple but illustrative example raised at the workshop is the "View-Source" control that still exists in most web browsers, allowing a user open access to the "HTML and scripting source code" that underlies a visible web page. View-Source was an early innovation that exploited the philosophy of openness, and provided a widely-used basis for learning how to write HTML, for enhancing web participation and publishing, for debugging web-pages, and for many validation activities. In this way it greatly helped to foster early generations of web content and to grow web activity to reach a critical mass.

***Advancement of science and CyberInfrastructure***

To make CyberInfrastructure a reality there will have to be massive software investment with significant risk and little evident near-term commercial payoff. There are strong prospects for OSS to increase the national software development capacity in general and in this area in particular. This is especially true in critical niche areas of science that will constitute the street-level CyberInfrastructure components and content. In many other areas of science OSS is already making significant, and sometimes dominant contributions to infrastructure and analysis. These impacts create strong motivations for better understanding the phenomena.

***Continuously operating systems with continuously evolving requirements***

Currently-dominant approaches to computer and information science and engineering don't recognize, embrace, or systematically examine continuous design concepts, approaches, processes, work practices, or socio-technical ecology. More importantly, many of the critical, globally pervasive components of the Web and CyberInfrastructure such as Apache Web servers, many Web browsers, Bind, Sendmail, OpenSSH, and operating systems including Linux and Free/Open-BSD clearly embody or depend on open source software. These cannot be shut down across the board for global redesign or replacement - instead, in aggregate, they are continuously operating in multiple interacting versions. The openness of development processes and code and the active community engagement mean that demand for change and evolution is also continuous, rather than punctuated. Users see and understand misfits, problems, and new opportunities continuously, not periodically.

In addition, for certain (e.g. open, continuously operating) systems, full requirements are in-principle unknowable in advance, and/or they change with significant frequency. The requirements for such software are fluid and continuously evolving rather than up-front, exhaustive and fixed. Such systems need to be designed and redesigned quickly and on the fly---while they are operating. Examples (both open and closed-source) include the power grid, the internet, and the air-traffic system. Continuous open systems require continuous design, and this is certainly a dominant trend in large OSS projects as the substrates and environments of software-in-operation change. Such software must be sustained for reasons of cost and infrastructure security, and we don't understand how to do it in a continuous manner.

***Openness is fundamental to development***

Openness in some forms and degrees underpins all large-scale and/or community-based software development efforts. Scalable software development depends on openness because of the need to coordinate collective efforts. Large-scale coordination requires information sharing; closed, sharing-inhibited development processes lead to participants either making too many unfounded assumptions in the absence of knowledge, or exerting too much control overhead to maintain levels of common ground that meet all constraints on information dissemination.

Several more general motivations are also found in the final section below.

## *Areas/topics needing further research*

After a total of three days of discussion among workshop participants, many important research problem areas/topics were identified. Space constraints in this preliminary report limit their presentation here. The following represent an unordered sample that cuts across the range and diversity of research interests in the continuous design of free/open source software, and were rated as highly important by workshop participants.

- How is the continuous design of free/open source software different from traditional approaches to design and engineering in research or commercial software product development venues? And to what consequence? What are the fundamental capabilities and limitations of continuous OSS development styles and processes?
- What kinds of software tools and system or component architectures work best with what types of OSS development? Generally, how does OSS community organization impact resulting software architecture, component structures, artifact quality/usability, and vice versa?
- How do large-scale communities understand, establish, coordinate, and evaluate the quality of the requirements and designs of continuous free/open source software? How are "informal" requirement and design representations coordinated and used?
- What design and use information is critical to capture and how can it best be organized for effective, continuous OSS development? How do knowledge, information, source code, data, and design artifacts migrate through and around free/open source software communities?
- How do participants learn about new project developments and how are new participants educated and brought into the process? (E.g., how could more HCI and usability expertise be migrated into OSS projects?)
- How are design processes and degrees/styles of openness related to the ability to create trustworthy, high-confidence systems?

- How scalable and sustainable are free/open source software communities, artifacts, design representations, and continuous design processes? How might they be made more scalable and sustainable?
- (How) can we systematically identify, collect, and comparatively analyze "great" designs, designers, and design processes as exemplars and prototypical cases?
- What are the best practices, significant examples, and critical success factors that result from or enable the continuous design of free/open source software systems and projects?
- What modes of discourse, native conceptual systems, and values do free/open source software developers use to characterize their designs and design practices?
- What are the most effective and efficient ways to model, visualize, and simulate free/open source software design processes, work practices, or community dynamics?
- How can continuous design methods for free/open source software best be incorporated into undergraduate and graduate CISE education? (How) can this improve education in these areas? What are the most effective practices for this?
- What policies should guide the acquisition, adoption, or use of free/open source software in academic, industrial, or government enterprises, and should these policies be continuously designed to respond to or anticipate national or international market conditions?

## *Research collaborations, shared research infrastructure/test-beds*

The workshop identified four kinds of research investment to facilitate research collaboration and shared infrastructure.

First, workshop participants believe the areas continuous design and free/open source software will benefit most from support of a diverse group of small to medium sized research projects. Because of the novel, emerging nature of the issues, it makes most sense at this time to explore the research space---we don't have enough knowledge in these emerging areas to define specific topics in which to make large investments with low risk. In general, large research projects or research centers are not a top priority national need at this time, with one clear exception: the creation and support of shared research and data infrastructures.

Second, to facilitate and encourage the emerging research community in this area, there is need for lightweight coordinating infrastructure components that can help build and sustain collaboration and mutual awareness within the research community. The idea is to support and sustain the development and use of community-based Web portal and communication systems such as threaded email discussion lists; blogs and wikis (open-ended collaborative authoring systems); and content management systems, that foster rapid dissemination of ideas, data, preliminary results, community events, and research findings.

Third, there is wide recognition of the need for shared infrastructure for the collection and management of data from non-profit and for-profit free/open source software repositories and portals. The SourceForge.net Web portal hosts information on tens of thousands of free/open source software projects, with similar numbers of developers, and ten times that number of registered users. Other portals like Freshmeat, Savannah, Tigris, Apache, Eclipse, NetBeans, Gelato and others host more narrowly defined and more specialized software, community, and project tracking information. Workshop participants expressed both strong interest in sharing data from some of these sources to which they already have access (including data from closed commercial software development projects), and strong interest in gaining direct, bulk access to data collected in such publicly-available portals. Done systematically, this gathering and sharing will require developing open interfaces (or application program interfaces/APIs) to data portals, as well as cleaning, anonymizing, and normalizing the data into standard representational formats. This activity represents investments that no individual research project can reasonably afford. More importantly, these potential data sets are a kind of national research treasure that is under-utilized. A core research investment to create a shared infrastructure that collects and manages secure, privacy-maintaining access to such critical data sets is a vital community-wide research need.

Last, free/open source software projects and communities are an emerging venue for very large scale design collaboration. In some projects/communities, thousands of software developers and users actively participate on an ad hoc, patterned, or routine basis in the continuous design of large software systems, and their associated development artifacts, processes, work practices, and community formations. For example, the community around the Apache Software Foundation (including the Apache Web server software and 40 or so other projects) now actively incubates small but potentially strategic open source software projects, so that they can grow into large, self-sustaining project communities. Such phenomena can be empirically investigated through both field studies as well as through experimental research efforts. Specifically, experimental efforts to create socio-technical infrastructures for designing, creating, and "cloning" online interactive communities for very large scale design collaboration is likely to be a core effort of the new CyberInfrastructure research program, and it is likely that such effort will have its roots in the technical information systems and collaborative community action found in free/open source software community.

## *Required investment*

The workshop participants assumed that individual research projects will be funded on average at $250K/year for up to five years, with some smaller and some larger projects. Similarly, they estimated that fifteen to twenty five research projects might reasonably succeed in being proposed, reviewed and recommended for funding in this problem area, and considered the potential growth of the research community working in this area. Given these assumptions, a research investment of $5M-$10M/year would realistically support 15-25 research projects in the first year ($5M/yr.), and could grow annually to eventually support 30-50 projects by the fifth year ($10M/yr.).

Beyond this, participants recognized that it was unclear whether to expect this level of funding to be organized within a single NSF program office, as a way to encourage program coherence and interdisciplinary research collaborations, or whether it might span several programs, and spreading programmatic risk and responsibility. However, participants agreed that a single program focus would encourage and stimulate the growth and cohesiveness of the research community.

## *Other recommendations for action*

Workshop participants identified a number of research issues that implicate or directly benefit from collaborations with industry, government agencies, and other academic researchers, and recommended the following actions in this regard.

First, there is growing interest and investment in the design and development of free/open source software in industrial contexts. Science and technology-intensive companies like IBM, SUN Microsystems, Hewlett-Packard, and Microsoft Research (MSR) are actively sponsoring open source software R&D projects (e.g., IBM-Eclipse, SUN-NetBeans and OpenOffice.org, HP-Gelato, MSR-Rotor) that involve not only salaried employees assigned to the projects, but also volunteers worldwide including academic researchers and students. (It's important to note that this is actual industrial OSS development work, and typically not research projects!) Whether and how these companies may benefit from the continuous design of open source software in the development of new products or services is unclear. However, participants from these firms indicated they are predisposed to either build on NSF research through co-sponsorship, engage academic collaborators, provide access to academic researchers whose research might be supported by NSF, and provide data access to research projects. Thus, NSF's research investment in this area may be complemented by industrial investments that expand the scope, depth, participation, and practical consequence of the research, and this should be actively encouraged and leveraged.

Second, many government agencies themselves are likely to benefit from a research investment in this area. NSF's programs in areas such as Digital Government, Bioinformatics and Genomics, and a National Virtual Observatory (in astrophysics) seem to be likely candidates to collaborate in supporting research on continuous design and development of free/open source components/applications for E-Government and E-Science. The CyberInfrastructure Program will be both a producer and consumer of free/open source software that emerges from continuous design and deployment efforts. NSF's investment in Education and Human Resources is likely to benefit also from the production and consumption of continuously designed free/open source software/content for formal/informal science, engineering and mathematics education, and from research into educational processes in the OSS community. Last, research in homeland security will need to systematically investigate whether and how the freedom and openness associated with the continuous design of free/open source software facilitates, inhibits or is irrelevant to cyber-security, as well as to the development and trustworthiness of government-based information systems.

Last, NSF is also positioned to stimulate research collaborations in the science of continuous design of free/open source software in international arenas. A small but growing proportion of free/open source software, particularly those emphasizing user interface design, user-user interaction, and public information services (e.g., design and access for digital libraries or digital government applications) are among the first to adopt user interface localization or internationalization. This localization enables further technology transfer and diffusion of results, as well as enabling new international research collaborations. For example, UNESCO has been supporting OSS projects developing digital library infrastructures and content, e.g. for rapid dissemination of knowledge for development, and for applications in preserving the indigenous information of native cultures that are disappearing or losing touch with their historical legacies. We also specifically encourage NSF to consider how to facilitate international research collaborations in the continuous design of free/open source software with scientists and engineers in China and India. China and India are recognized as nations anticipating substantial development, adoption, and proliferation of free/open source software. Both countries have a sizable technically skilled labor force, large populations, and nascent or growing indigenous IT industries. Whether widespread adoption and proliferation of free/open source software technology in these particular countries favors or hampers U.S. interests in unclear, but it merits careful study that can be fostered through international collaborations.