Building Networks of Software Communities in a Large Corporation

Catharina Melian[1], Cathy Burles Ammirati [2], Pankaj Garg, Guje Sevon [3]
Software Technology Laboratory
HP Laboratories Palo Alto
HPL-2002-12
January 23rd , 2002*

E-mail: cme@fek.su.se, cathy_ammirati@hp.com, garg@hpl.hp.com, guje.sevon@hhs.se

In this paper we report on an ongoing research to understand and foster networks of communities around the development of software systems in a large corporation. Software systems are integral to the functions of any modern large, corporation. Such systems minimally include word processors, email, group communication systems, and often include complex systems such as Enterprise Resource Planning (ERP), Enterprise Portals, and so forth. Hence, most modern corporations of today heavily utilize and often develop software systems or their customizations. The success of several Open Source software systems, e.g., Linux, Apache, and so forth, leads us to believe that certain collaboration practices of Open Source development methods, like open discussions for features and requirements, the ability of the user community to participate in such discussions with the developer community, will benefit large corporate software development and customizations. In particular, we aim to document the ongoing efforts within Hewlett-Packard, to replicate the collaboration styles of Open Source development methods within a corporate setting. We report on the development of "Progressive Open Source," as defined at Hewlett-Packard, and the nature of network communities developed to support it. We formulate and describe a study aimed at describing the fostering of such networked communities within Hewlett-Packard. While the study is in progress, early results indicate the positive influence of Progressive Open Source in creating diverse networking groups within the company.

* Internal Accession Date Only
Approved for External Publication
[1] Stockholm University School of Business, Sweden
[2] HP Vancouver, WA, USA
[3] Stockholm University School of Economics, Sweden

1. Introduction

Today, companies that previously developed software in a closed environment, i.e., without revealing the source code[5] of their software, are acknowledging the advantages of the open environment, and seek solutions leveraging the advantages of the "Open Source" movement. Tools and methods emanating from the Open Source community are developed and reconstructed in order to suit the needs of large corporations. An important issue is obviously to protect Intellectual Property Rights, while at the same time sharing and exchanging information as well as knowledge and experiences e.g., source code and other related information (Lessig 1999). An entire new paradigm, the **Progressive Open Source** is currently influencing practices, aimed at rejuvenating software engineering through combining existing practices with tools and technology widely used and utilized within the community of Open Source developers. Progressive Open Source sets out to establish a set of centralized techniques, tools and infrastructure enabling fast, convenient and effective communication between developers and third parties engaged in short and medium duration projects. Interoperability is sustained by standard protocols and syntaxes. Distributed collaboration is dependent upon global access, common solutions and tools, and standards. The idea of involving third parties and the customer in the process of innovation is important, since it is a way of anticipating the need, problems, and competence of the other actors. Victor and Boynton assert that co-configuration, i.e., development in conjunction with the customer, increases adaptability and learning capacity, as well as strengthening the relationship with the customer (1998). Following Latour, artifacts and network of actors are co-constructed in the process of innovation indicating a need to further investigate how collaborative relations between actors in a software development network are constructed (1987, 1999). Individuals participate with their actions in dynamic networks, and by studying the individuals we can say something about the network.

Traditional theory on co-operative and collaborative efforts is mainly built and derived from studies of companies in a resource-based setting, i.e., subjected to the laws of diminishing returns. Over the last decades, however, companies are increasingly becoming knowledge-based, and by corollary subjected to the laws of increasing returns. Computers and software programs are complex and expensive to manufacture and sell. But once invented, the incremental production is relatively cheap. As more products are built the costs continue to fall, and profit increases. Moreover, knowledge does not disappear when used, but it can be used over and over again, i.e., a learning economy is characterized by a net gain in knowledge. Why is that important then? Economists have argued that the pooling of resources is a viable and effective way for firms to compete, since up-front costs, marketing networks, technical knowledge, and standards may be shared (Arthur 1994). When exploring the pressures forcing companies to change we have to take into account what used to work then (in the past) and what seems to be working now (and possibly in the future). It has been suggested that companies are dancing on a tight rope between extremes, e.g., (Mauss 1924/1990, Kollock & Smith 1999, Barbrook 1999 Rehn 2001)

- A commodity driven economy versus a gift economy
- Proprietary, closed software versus Open Source systems
- Market competition versus network communities
- Digital encryption versus free downloads

[5] Source code is basically the running program of a system, i.e. the underlying code that constitutes the software program. Closed or "black-box" systems are typically written in binary form that makes them incomprehensible, (Neumann 2000)

As companies try to adapt to changing conditions, hybrid solutions emerge. The Progressive Open Source is worth exploring since it is an emerging hybrid, representing ongoing changes in large companies.

We report on an ongoing study that explores how it is possible to foster communities of practice supported by software technologies within a large corporation, i.e., creating and sustaining a viable knowledge network, embracing third party engagement and open source contributors. We seek to analyze why it is difficult to do so today, i.e., what seems to be impeding effective collaboration and how the problematic issues may be resolved in the future.

1.1 Collaboration for Technical Innovation

Research reported by Dinkelacker, Garg, Miller & Nelson indicates that software engineering continues to impose challenges for large corporation (2001). While at the same time, several studies have evaluated and explored the successes within Open Source software systems, e.g., Apache, Bind, Emacs, and Linux (Tuomi 2001, Raymond 2001, Jae Yun Moon & Sproull 2000, DiBona, Ockman & Stone 1999, Wayner 2000, Moody 2001). The studies show that a large community of contributors typically develop Open Source systems in a joint effort facilitated by communication through electronic newsgroups and mailing lists on the Internet. Many Open Source systems seem to have progressed with voluntary resources, i.e., without assigning work to specific individuals and without an explicit system-level design activity(Mockus, Fielding & Herbsleb 2000). In fact, the successes of the Web and Linux indicate that it's more efficient and competitive to support development in an "Open Standard" way, i.e., any protocols or software interfaces are discussed in an open discussion forumn with both the users and developers actively participating or monitoring the discussion. Once the "standard" has been defined in this manner, anyone can build supporting software tools to work off that standard. For example, once the HTTP protocol was defined by such open discussions, different groups of software developers could develop the clients and servers independently to complete the development of the World Wide Web.

Large companies are increasingly seeking to leverage from their capacity to generate knowledge while processing as well as managing information. The scope of the activities is also expanding to include a selective but yet global workforce capable of working on a planetary scale in real time (Castells 2000). The enterprise is furthermore networked in the sense that corporations chose a strategy of changing alliances and partnerships for different projects, thus seeking to speed up the process of innovation and overall performance.

Innovation is considered to be a major competitive advantage for business and organizations, and extensive empirical research indicates that innovation is dependent upon the development and management of knowledge (Christensen 1997, Zeleny 1989, Nonaka 1994, Sevón and Kreiner 1998). Management techniques, however, have yet to be further developed in order to activate and manage productive cooperation and collaboration supporting a smooth interface between different functions in the organization, between different work teams, and most importantly between different levels of the hierarchical organization. The omnipresence of technology may potentially enable actors to network, but social and cultural issues remain at least equally important.

In the following section we discuss Progressive Open Source and its implications for a large corporation. We proceed with an exposition of some of the organizational challenges that have been acknowledged so far. Thereafter we give a description of an ongoing case study at Hewlett-Packard that seeks to address issues of fostering global networks of software communities. The article is concluded with some tentative results and usage statistics.

2. Progressive Open Source

Dinckelacker et al. (2001) define "Progressive Open Source" as a strategy for large corporations to adopt Open Source software development methods. In essence the concept encompasses coordinated resource sharing and problem solving in a dynamic, multi-actor virtual network. The hypothesis is that by adopting Open Source development methods within a corporation, the corporation can gain from the collaboration styles of the Open Source software methodology resulting in robust code quality, features that are well-tuned to user's requirements, strong, well-established networks of communities of practice, and so forth (Brown & Duguid 1991, Chaiklin & Lave 1993, Lave & Wenger 1991). Progressive Open Source (**POS**) advocates the progressive adoption of Open Source practices by a corporation in primarily three stages:

- **Inner Source**: open the software source code to only employees of the company,
- **Controlled Source**: selectively open source code to third parties and partners, and
- **Open Source:** open source code to the entire Internet community.

Each one of these stages results in its own network of communities of practice. Any given software project within the corporation can choose to participate in one or more of the three communities. Given the common starting point of the corporation, however, each community will eventually benefit from, and utilize, the networks of the other two communities through cross-linkages and common members.

The three identifiable communities of POS are: (1) the internal communities of software developers within the company, (2) the communities of people mixed from the company and its partners, and (3) the Open Source software development communities. The third kind of community, the Open Source community, has recently received much attention from researchers (cf. Mockus et al., 2001), Moon and Sproul, 2000). In this work, we are more interested on understanding the first two kinds of communities, i.e., communities of practice around software systems within a corporation and the corporate partners and third party relationships. In HP, we have defined two programs to leverage such novel communities: (1) *Corporate Source*, and (2) *Collaborative Development Program (CDP).*

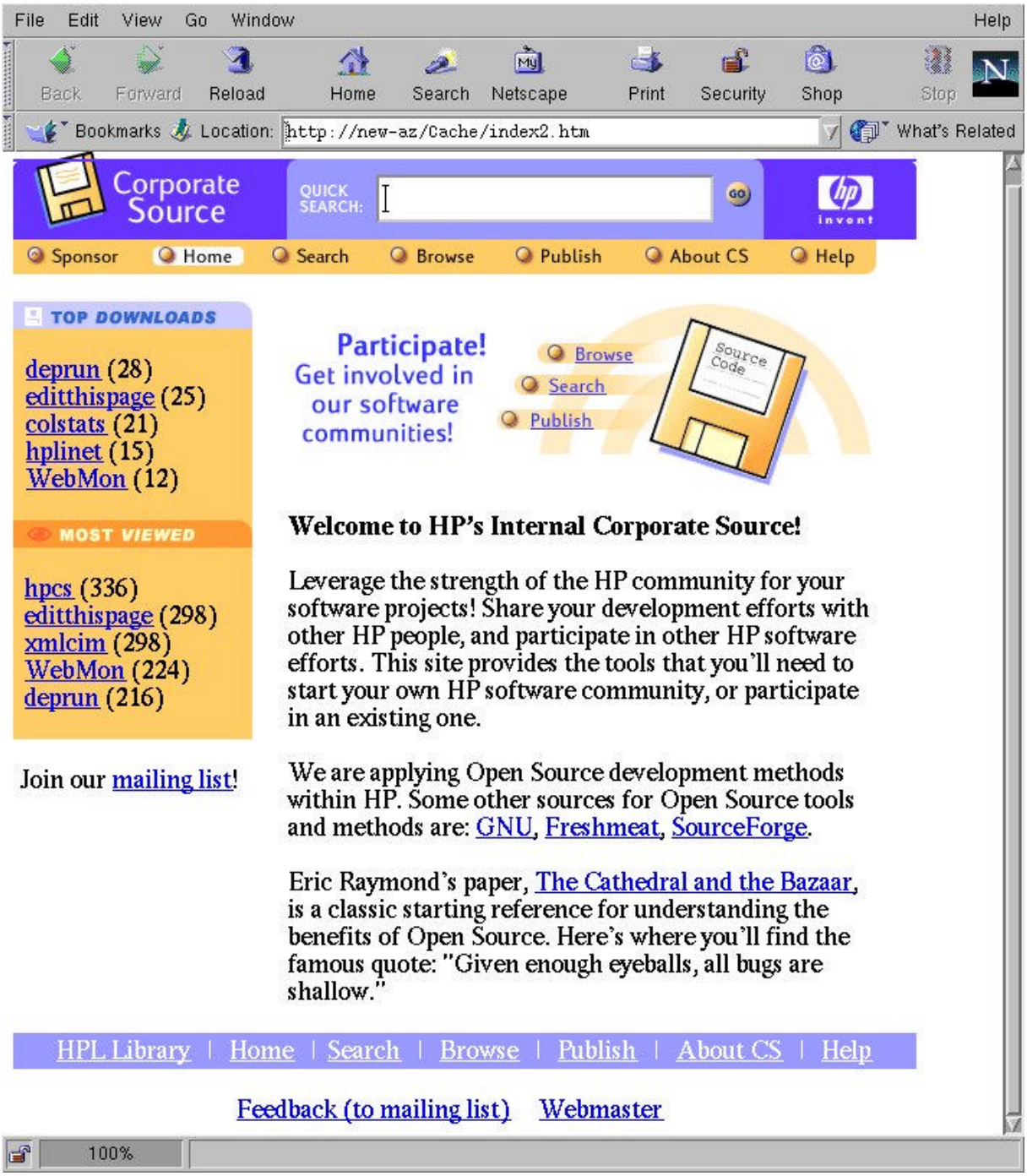
Corporate Source advocates the use of flat, networked organizations for software development in large corporations. To explain the idea, we contrast it with the current, hierarchical form of organization for most software projects. In companies like HP, software products are organized in hierarchies, either functionally or market-driven. Hence, a product group in the printer division writes all printer software, and the operating system group in the computer systems organizations writes the operating system software. The only connection between these two groups is through the Chief Executive Officer, who is often up to ten or twenty levels higher than the engineering groups working on the product. The source code of the software from one group is rarely available to the other group. Hence, if there is a problem in the interface between the printer driver and the spooler on the operating system, several layers and channels of communication and coordination have to be crossed to address the issues. In contrast, Corporate Source advocates that the two groups (and all other groups in the corporation) should freely make available their source code among themselves. In this manner, the printer group should be able to make changes to the operating system spooler source code, and the operating system group should be able to modify the printer driver source code. The ownership and control of what ultimately goes in the product still rests with the original owners.

Corporate Source borrows heavily from the Open Source development paradigm (DiBona et al., 1999) and from the methods of scientific research (Kuhn, 1970). From the *Open Source development paradigm*, it borrows the notion of making source code available freely (openly) for all members of the community; advocates the use of open email discussions for feature addition, implementation, review and testing; and provides a persistence base for the source code and email discussions to be available long after their creation date. In this manner, a new person can quickly join a project by understanding the rationale behind some feature selection and implementation (Raymond, 2001). We utilize the World Wide Web (WWW) infrastructure to make Corporate Source projects freely available for all employees to browse through and participate, through a familiar employee portal. From the *scientific methods*, Corporate Source borrows the notion of "publishing" work for peer-review and criticism, and archival storage of important experimental results for future review. Hence, the primary responsibility of facilitating the use of Corporate Source rests with HP's research library, which is also the primarily responsible for maintaining and disseminating HP's scientific knowledge as technical reports.

The HP Research Library hosts the Corporate Source service, along with some other knowledge management services, such as a database of skill set of employees, an Idea forum, technical reports, and so forth. Figure 1 shows a typical screen of the Corporate Source service that provides a community hub for the users of Corporate Source. Members of the community can publish their own software, update an existing software, search or browse through existing software, or comment, criticize or review existing software or discussions. Traditional hierarchical organizational boundaries are minimized by only exposing relevant information about a user's network identity and skill set. Any given user's hierarchical position can, of course, be determined through some of the attributes of the network identity. Hence, one cannot truly achieve a virtual network identity as in the case of the Open Source development, where any given user can completely hide behind a network identity. The nature of the discussions and the corresponding contributions to the software, ideas, and thoughts, therefore, may be different in Corporate Source than in the Open Source communities. We are looking at ways of studying such behavior in order to effectively mitigate the loss of value from artificial hierarchical boundaries of current organizations. We can then truly realize the potential of Corporate Source by empowering even the junior-most member of the organization (who may not be professionally trained as a software engineer) to make far-reaching and wide contributions to the corporation's software, similar to what we see in the Open Source communities when high-school graduates are able to shake up the software and media industry, for example with the work of peer-to-peer computing of Napster[6] (Clarke, Sandberg, Wiley & Hong 1999, Alderman 2001, Rose & Buchanan 2001).

[6] Napster is a protocol for sharing files between users. See e.g., http://opennap.sourceforge.net/

Figure 1: Main screen for the Corporate Source service



The **Collaborative Development Program (CDP)** is advocated by HP's Imaging and Printing group that develops the printers and related products for HP. Several printer products have relationships with each other and third party products, e.g., the all-in-one office jet has some features that are in common with a printer, while other features that are in common with a scanner, and so forth. The source code for the firmware and software for these features, therefore, must be shared between the groups that provide these products. Similarly, software for some of the foundational features, like networking, is common to all the groups. The goal of CDP is to foster appropriate networking,

collaboration, and community spirit among the various groups that participate in such development, including third party individuals who may not be employees of HP.

While the Corporate Source program is a grass-roots program from HP Labs, CDP is a group funded and organized program that has executive champions, sponsors, R&D change leaders, and information technology staff. The executive champions have been critical in establishing the credibility of the program and establishing a need for collaboration among the different groups in the organization. The sponsors are critical in committing resources that enable short-term and long-term nurturing and success of the program. The R&D change leaders enable a dialogue of cultural change and training in the organizations to begin the long process of transition from a hierarchical, product-focused organization to a networked, collaboration-focused organization. The IT staff provides the critical collaboration infrastructure on a 24x7 supported basis. The CDP infrastructure supports email discussions and bulletin boards, source code repository (searchable and browsable), and defect tracking. While the Corporate Source infrastructure is a home-grown combination of some available Open Source tools like CVS (Fogel 1999) and Mailman (http://www.gnu.org/software/mailman/mailman.html), CDP relies on a third-party, CollabNet, Inc., to provide the tools infrastructure.

The CDP infrastructure resides on the Internet (as opposed to the Intranet for Corporate Source). Hence, bringing in a new third party on-board the CDP infrastructure is a relatively straightforward task. Indeed, the time for establishing a new collaboration project can be reduced to a matter of few days from what it used to be a few months or weeks at best. If all the project participants are from HP, then the setup takes a few hours. Each user in CDP is given a network identity, which is based on their corporate identity. The hierarchical organizational identity, however, is not that easily visible although it can be deduced quite easily. CDP organizes people by projects and by default any HP employee is given read-access to any "open" project in CDP. CDP promotes the sharing of knowledge and information to build a community that will deliver on the priorities across the company. Ultimately the goal is to do away with any organizational boundaries to allow engineers to apply their expertise to provide the greatest return for the company by enabling project teams to deliver innovative solutions faster and with greater reliability.

Both Corporate Source and CDP have been operational in HP for a several months. Corporate Source was officially launched in June 2000, and has been operational for eighteen months. CDP was launched in April 2001, and has been operational for about eight months. Corporate Source has about 1500 registered users while CDP has 3000 users (10% of whom are non-HP). Forty-five external companies are developing projects with HP using CDP. Corporate Source has about two dozen projects, all of which are research projects that are not tied to any HP product. CDP has about 350 projects, most of which are tied to specific HP products. Corporate Source has users in forty-five countries; CDP has users from at least eighteen countries. Both community hubs have active users, although we need to work on increasing the awareness, adoption, and use of Progressive Open Source within HP. Along with providing the right set of tools, we must provide the appropriate organizational structures, rewards, and motivations to transition HP software development into a more collaborative, Open Source style of development. We next describe some of the organizational challenges that are still to be effectively addressed.

2.1 Organizational Challenges

Progressive Open Source is a fundamental shift in the software process and the supporting organizational and technology infrastructures. Modern organizational structures have continued to rely

on the hierarchy for efficient communication and coordination, since hierarchies require less information exchange among their parts than do teams (Simon 1977). Hierarchies also tend to be more robust in terms of resisting communication errors (Carley & Lin 1995) Software development is typically considered as unstructured and non-routine tasks, by corollary it relies on a large extent of informal interaction for coordination, (Kraut & Streeter 1995, Van de ven, Delbecq & Koening 1976)

Establishing a parallel, networked, virtual organization within a hierarchical organization to support POS is the primary challenge for an effective implementation of POS (e.g., Davidow & Malone 1993, Yakhlef 2002). We define virtual organizations as groups of people working and interacting across time and space constraints and organizational boundaries, facilitated by webs of communication technologies (Lipnack and Stamps 1997). Progressive Open Source is a paradigm to support a virtual research organization, in which people from different units and from separate organizations interact through interdependent tasks, guided by a common purpose, i.e., to advance a technology on an ongoing basis. Successfully mirroring the Open Source movement, which was guided by a multitude of minute, individual forces over a long period of time, within a corporate hierarchy in a time-critical market, is an aspect of this challenge. More specifically, we face the following organizational challenges for a successful adoption of POS within HP:

Leadership: The Open Source approach relies on the leadership of an individual, or a few small numbers of individuals, to guide the design, implementation, and discussion forums in a software community. In the Open Source world, such leaders arise for some communities, and those communities thrive. For other communities, no one ever emerges as a good leader and their communities disappear or languish. In the free world of Open Source, the "market" does not necessarily miss such communities as closed-source businesses rush in to fill the void and make a business from selling software that is not available in a robust form from the Open Source communities. For a business corporation, however, such a model is unacceptable. The leaders must be present in the corporation at appropriate times to undertake the leadership of the appropriate community and lead it to success. Hence, corporations must hire individuals with the potential to establish such networked communities.

Collaborative Work Style: The traditional hierarchical organizations reward and promote cohesive project or product related behavior. For example, in HP, individuals are evaluated every year on their contributions to their assigned project. While community-help and visibility is encouraged, it is not the main factor when considering the yearly progress of an employee or their managers. Helping out another person in a different group can sometimes be detrimental to an individual's career. The other group or individual may be more successful based on that help, and eventually in the fight for limited resources within the corporation, may indeed go against the original good Samaritan. A multi-party collaboration, therefore, that benefits all parties concerned must therefore be established, motivated, and rewarded. Individuals must be able to understand where its in their and their group's best interest to collaborate and what actions to avoid that can potentially lead to giving strength to their corporate competition.

Task Assignment: In the Open Source community, individuals contribute to a given project because they have an unmet requirement and have the skill set to accomplish the required software coding on their own (Raymond 2001). On the other hand, assigning specific tasks to hired employees develops corporate products. Suppose the skill set and requirements of an employee, E match for a given software project P. How can the employee E justify spending time on project P. Moreover, how can the manager of project P quickly gain managerial supervision of E to accomplish the critical

requirements of project P? Current hierarchical organization places strong barriers against such movements and task assignments.

3.0 Research Approach

In this section we describe our approach for understanding the adoption and use of Progressive Open Source within HP. We describe what we seek to explore, and how we intend to collect the empirical data.

The research focuses on collaboration as a mean to improve and speed up innovation in software engineering. We seek to explore what collaboration is all about in practice, i.e., how individual software developers perceive collaboration and collaborative efforts, what it is and what the results are. Organizing for innovation is crucial for companies that are forced to quickly adapt to changing competition, markets and technologies (Dougherty 1996, Hage 1988, Jelinek and Schoonhoven 1990, Zahra and Covin 1995). Despite this awareness, activities that are directed towards changing established structures in the organization and encourage innovation continues to be challenging for the organizations. Collaboration in itself has an ambiguous connotation. It embraces positive and encouraging aspects such as the willingness to work jointly on intellectual endeavors. But also less positive aspects such as control, visibility and giving things away, possibly to an enemy (translated into a business setting: a competitor). This inherent tension must be mitigated when organizing communities of developers within a corporate setting. Other tensions are obviously the market-technology linking, i.e., keeping operations efficient inside while maintaining the relationship with third parties and customers. Organizing for creative problem solving highlights the balance between effective reuse of already invented ideas while at the same time making room for new ideas. Collaboration also needs to balance between the individual and the collective, e.g., in regard to developing a commitment to innovation, shared responsibilities and accountability. We believe that ethnographic studies of technology-mediated collaborative work can provide important insights in the social interaction, e.g., how work is coordinated and how unexpected events are handled and co-managed (Heath & Luff 1991, Goodwin & Goodwin 1996, Suchman & Trigg 1991, Bentley et al 1992)

Our nature lies in movement: complete calm is death.[7]

Czarniawska and Sevón advocate that researchers on the field have to understand ongoing changes through observing the events as they are unraveling, e.g., by listening to the testimonies of those affected (1996). It is important to study changes since "only what moves is visible" (1996:2). In times of changes old things and habits are put to sleep and new ways are brought to life, they are *constructed*. We try to collect 'little narratives' in search of understanding, striving to capture the richness of the meaning of collaborative action, evoking the associations with movement as well as association (Czarniawska & Sevón 1996). At the end of our project we wish to be able to say something about how local action emerges and how it becomes institutionalized within a large corporation. The only way to do this is to talk to those who know, i.e., the organizational actors.

The goal of the research is to create a multi-viewpoint understanding of the new software paradigm, Progressive Open Source. We set out to do so by combining different research methodologies and theories. The research design encompass participatory observations, semi-structured interviews with users of the Progressive Open Source, as well as quantitative analysis of data collected on the community use of the collaborative systems, i.e., how the technology is utilized within the community

[7] Blaise Pascal, cited in Bruce Chatwin 1988:183

and how the community grows and proliferates. We will conduct in-depth interviews with selected users of Corporate Source and the Collaborative Development Program. The interviews will cover the spectrum of managers, Human Resource Officers and engineers. For this part of the investigation, we have developed a framework of questions. The investigation aims at providing insights on the behavior of the adopters of the new methodologies offered by Progressive Open Source within Hewlett-Packard. The user statistics are reviewed in order to define and categorize the adopters. We will focus on particular projects including third party collaborators. In order to reduce biases and ensure confidentiality, we protect the integrity of the interviewees, i.e., they participate on an anonymous basis.

The study is expected to contribute to a comprehensive and deeper understanding of how the Progressive Open Source paradigm is received by the HP development community. In addition, we address similar questions to managers and Human Resource officers to grasp how they perceive Corporate Source and the Collaborative Development Program and its importance for innovation. The study addresses eight main areas, even though the interview design allows deviations in terms of addressing the issues that are perceived as important by the users. We now briefly describe the areas and what we want to explore.

By way of introduction it is important to understand what the developers perceive as innovative about the projects facilitated by Progressive Open Source. Furthermore, what the impact is on their day-to-day work. We wish to know more about the advantages respective disadvantages from an individual as well as a team perspective.

Also, we are keen to collect the success stories that have already occurred as a result of using Progressive Open Source. Moreover, we are interested to know more about the characteristics of innovation e.g., the relative advantage, the compatibility, complexity, trialability and observability (Rogers 1962, 1995). Another area of interest is motivation and how collaborative environments may be encouraged and how individuals and teams are to receive sufficient rewards and recognitions for their contributions. We draw on the work of Mauss (1924/1990) to try to understand more about gift-cultures and how people are motivated to make contributions to a community. Existing reward structures within the company may conflict with the notion of the gift economy. Finally, we address issues directly related to third party interaction e.g., communication, support, security and trust, and most importantly strategy and issues of corporate philosophy.

In addition to the above-mentioned interviews, we collect stories from the field in order to grasp the context of Progressive Open Source, i.e. where it comes from and why it is perceived as important by a large corporation to adopt methodologies and ideas from the Open Source model of developing Software. The research is best described as being inter-disciplinary, since the methodologies stem out of the software-engineering paradigm as well as from organizational studies. The research project is designed and undertaken as part of a doctoral thesis in business studies in close conjunction with researchers on software engineering within a corporate entity, Hewlett-Packard Labs in California.

4.0 Expected Results

Openness is part of HP culture. Early on, the founders of HP, Bill Hewlett and David Packard, understood and promoted an “open-door” policy in HP. The basic theme was that there are no doors on anyone's office, and anyone can communicate with anyone, regardless of hierarchical positioning (Packard 1995). Clearly, collaboration is just as much an issue of culture and tradition as it is technology. The Progressive Open Source has evolved out of a context rich of precursors and

prototypes of collaborative development models utilized within the company. We wish to highlight one of them that exhibits most of the traits acknowledged in the collaborative development model applied today, the Owen Firmware Cooperative model (Toft, Coleman and Ohta, 2000).

The Owen Firmware Cooperative project stems out of an incessant business constraint, leading to an emergent need to do more, preferably better, with the same resources and investments. Before the Owen project, a few related products were developed serially. In order to meet business goals (time-to-market) several related Owen products were developed in parallel through enhanced reuse and development efficiency. A technical approach was adopted in order to create a firmware-architecture and to define and adopt common tools in order to facilitate code sharing and leverage. In fact, it was acknowledged that the code in itself is valuable and important, not only for the product being developed, but for future products too. In order for other developers to be able to pick up the code, they must be able to understand and make use of it with ease. This was feasible through well-defined interfaces, i.e., interface descriptions represented in a standard form. Moreover, it was stressed that the components must be loosely coupled, and the establishment of clear principles and guidelines for using and extending the architecture. Above all, cooperation and collaboration was guiding the work.

The Owen project also indicates that it is of great importance to establish and build strong working relationships and trust, especially when the work teams are globally distributed. However, it was also acknowledged that contribution to the community is far more important than distribution of work within a global community. A set of operating principles were confirmed in order to stipulate the important features e.g., leverage of existing knowledge and progress, critical but simple guidelines, the importance of contributions, project team autonomy and empowerment, and a focus on maximizing utility of the contributions.

The Owen project resembles Open Source Projects in several ways: its emphasis on empowerment, and stress on the importance of maximizing the utility brought back to the community of users. The management structure differs from the traditional, e.g., power and influence in the development process, is guided by competence, rather than formal status (meritocracy). The code in itself is regarded as valuable, not only the end product. And finally, the projects carefully monitor and manage the evolution of interfaces.

The Owen project was a big success: prior to the Owen cooperative, All-in-One devices were introduced 18 months after the equivalent Single Function Printer, today All-in-One devices are co-introduced with the equivalent Single Function printer. The success of Owen gives us assurance that given the right organization and supporting infrastructure, Progressive Open Source will be successful in HP. The work reported in this paper is a step towards addressing the core organizational issues to be addressed for a successive implementation of Progressive Open Source.

Tentative results indicate that Progressive Open Source and its precursors facilitate collaborative efforts leading to improved conditions for software development, re-use and innovation within Hewlett-Packard. At this point the efforts are gaining global reach, user statistics indicate that Progressive Open source operates in at least 55 countries, in spite of lacking central project headquarters and national origins. New users and projects are being added to both the Corporate Source and CDP programs at a steady rate. For example, the following graph shows the addition of new users to the Corporate Source service on a monthly basis.

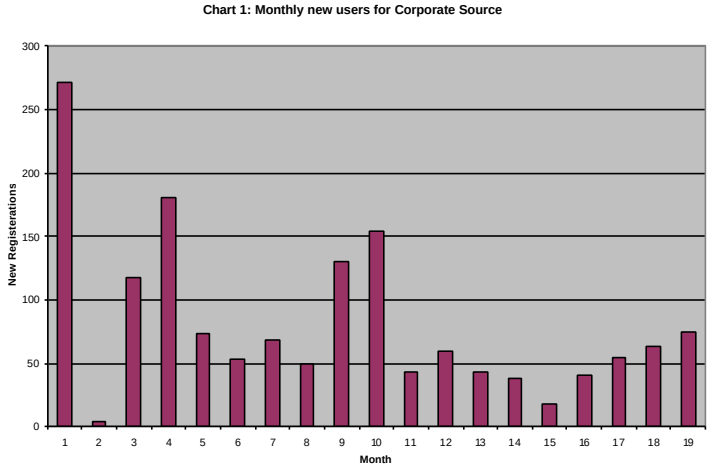
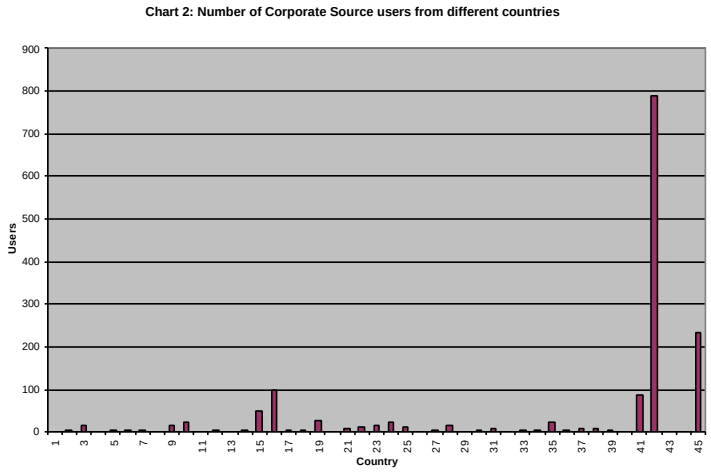


Chart 2 shows the number of registered users from different countries:



5.0 **Summary**

In this paper we described the use of Open Source collaboration style within a large company. We described an innovative program, called Progressive Open Source, that has been defined at HP to leverage the collaboration benefits of Open Source development and use methods. The program currently has a global reach, with no project headquarters, no national origin, and operates in at least 45 countries. Early experiences indicate increased location independence, i.e., developers can remain physically close to customers, while working wherever is convenient. Developers also are easily re-deployed between different projects. Progressive Open Source is also an electronic Knowledge Network that facilitates the decentralized organization. The computer networks support and supply

the actors with corporate information, that is available online, worldwide, for immediate access. Early statistics indicate time compression, i.e., development can be maintained and supported on a 24 by 7 basis, i.e., towards a 24-hour workday. Potentially, software projects never sleep – they follow the sun. Programmers working in different time zones ship code back and forth to keep the development process moving. The research reported in this paper constructs a theoretical framework to understand and document this new collaboration paradigm at HP.

References

Alderman, Johan, (2001), Sonic boom: Napster, MP3 and the new pioneers of music, London: Fourth Estate

Arthur, W. Brian, (1994), Increasing returns and path dependence in the economy, Ann Arbor University of Michigan Press

Barbrook, Richard, (1998), The High-Tech Gift Economy, First Monday 3.12, http://www.firstmonday.dk/issues/issue3_12/barbrook/index.html

Bentley, R., Hughes, J.,A., Randall, D., Rodden, T., Sawyer, P., Sommerville, I., Shapiro, D., (1992) Ethnographically-informed systems design for air traffic control. In proceeding of the Conference on Computer Supported Cooperative Work, CSCW 92, ACM, New York, pp. 123-129.

Brown, J., S., & Duguid, P., (1991), Organizational Learning and Communities-of-Practice: Towards a Unified View of Working, Learning, and Innovation, Organizational Science 2 (1): 40-570

Carley, K. & Lin, Z., (1995), Organizational designs suited to high performance under stress. IEEE – Systems, Man and Cybernetics, 25 (1), 221-23

Castells, Manuel, (2000), Materials for an exploratory theory of the network society, British Journal of Sociology, Vol. No. 51, January/March 2000, pp. 5-24, ISSN 0007 1315, London School of Economics.

Chaiklin, S., & Lave, J. (eds), (1993), Understanding Practice: Perspectives on activity and context, Cambridge: Cambridge University Press

Chatwin, B, (1988), The Songlines, London: Picador

Christensen, Clayton, M., (1997), The innovator's dilemma when new technologies cause great firms to fail,, Boston, Mass. Harvard Business School

Clarke, I., Sandberg, O., Wiley, B., & Hong, T.,W., (1999), Freenet: A Distributed Anonymous Information Storage and Retrieval System. In ICSI Workshop on Design Issues in Anonymity and Unobservability

Czarniawska, Barbara & Sevón, Guje, (1996), Translating Organizational Change, Berlin de Gruyter

Davidow, William, H., & Malone, Michael, S. (1993), The virtual corporation: structuring and revitalizing the corporation for the 21st century, New York, Harper Business

DiBona, Chris, Ockman, Sam & Stone, Mark, (1999), Open Sources: Voices from the Open Source Revolution, O'Reilly

Dinkelacker, Jamie, Garg, Pankaj, K., Miller, Rob, Nelson, Dean, (2001), Progressive Open Source, HP Laboratories Palo Alto, HPL-2001-233, Hewlett-Packard Company. To appear in the *Proceedings of the 24th International Conference on Software Engineering*, Buenos Aires, Argentina, May 2002.

Dougherty, Deborah, (1996), Organizing for Innovation, in Handbook of Organization Studies, edited by Clegg, Stewart, R., Hardy, Cynthia & Nord, Walter, R., Sage Publications

Fogel, Karl, 1999, Open Source Development with CVS, Cariolis, The United States

Gooewin, C., & Goodwin, M., J., (1996) Formulating Planes: Seeing as a Situated Activity. In Middleton, D & Engeström, Y (eds.) Communication and Cognition at Work: Cambridge University Press, Cambridge pp. 61-95

Hage, J. (ed), (1988), Futures of Organizations, Lexington, MA: Lexington Books

Heath, C., & Luff, P., (1991), Collaborative Activity and Technological Design: Task Coordination in London Underground Control Rooms, In Proceedings of the Second European Conference on Computer-Supported Cooperative Work, Kluwer, Dordrecht, pp: 65-80
Jelinek, M., & Schoonhoven, C., (1990), The Innovation Marathon: Lessons from High Technology Firms, Oxford: Basil Blackwell
Kraut, R., & Streeter, L. (1995), Coordination in Software Development, Communications of the ACM, 38 (3), 69-81
Kuhn, Thomas (1970), The Structure of Scientific Revolutions, Second Edition, Enlarged, The University of Chicago Press, Chicago, IL.
Latour, Bruno, (1987), Science in Action, Cambridge, MA: Harvard University Press
Latour, Bruno, (1999), Pandora's hope: essays on the reality of science studies, Cambridge: Harvard University Press
Lave, J. & Wenger, E., (1991), Situated learning: Legitimate peripheral participation, Cambridge: Cambridge University Press
Lessig, Lawrence, (1999), Code and other Laws of Cyberpace, Basic Books
Lipnack, J., & Stamps, J., (1997), Virtual Teams: Researching across space, time, and organizations with technology, New York: John Wiley and Sons.
Mauss, Marcel, (1924/1990), The Gift: The Form and Reason for Exchange in Archaic Societies, (transl. W.D. Halls), New York: W.W. Norton
Mockus, Audris, Fielding, Roy, T., & Herbleb, James, (2000) A Case Study of Open Source Software Development: The Apache Server, Bell Labs, 263 Shuman Blvd., Naperville, IL 60566 USA, Proceeding of the 22nd International Conference on Software Engineering, Limerick, Ireland, June 2000
Moody, Glyn, (2001), Rebel Code, Inside Linux and the Open Source revolution, Perseus Publishing, Cambridge, Massachusetts
Moon, Jae Yun & Sproull, Lee, (2000) Essence of Distributed Work: The Case of the Linux Kernel, http://firstmonday.org/issues/issue5-11/moon/index.html
Neumann, Peter, g., (2000), Robust Nonproprietary Software, IEEE Symposium on Security and Privacy, Oakland, CA May 15-17
Nonaka, Ikujiro, (1994), A dynamic theory of organizational knowledge creation, Organization Science, 5(1):14-37
Packard, David, (1995), The HP Way, How Bill Hewlett and I Built Our Company, Harper Collins, New York
Raymond, Eric, S., (2001), The Cathedral and the Bazaar: Revised Edition, O'Reilly, CA
Rehn, Alf, (2001), Electronic Potlatch, A study on new technologies and primitive economic behaviors, KTH INDEK, Doctoral Disseration
Rogers, Everett, M., (1962, 1995), Diffusion of Innovations, New York, Free Press
Rose, Nick & Buchanan, Nichola, (2001), A&M Records v Napster: the case and its consequences for copyright owners in the music industry, Copyright world 2001:112, 13-15
Sevón, Guje and Kreiner, (1998), Constructing R&D Collaboration, Copenhagen Business School Press
Simon, H., (1977), The new science of management decision, NJ: Prentice Hall
Smith, M., A. & Kollck, P., (eds) Communities in Cyberspace, London: Routhledge, 1999
Suchman, L. & Trigg, R., (1991), Understanding practice: video as a medium for reflection and design, In Greenbaum, J. & Kyng, M. (eds.), Design at Work: Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 65-90
Toft, Peter, Coleman, Derek, and Ohta, Joni, (2000), "A Cooperative Model for Cross-Divisional Product Development for a Software Product Line," in Patrick Donohoe, Ed., *Software Product Lines: Experiences and Research Directions*, Kluwer Academic Publishers

Tuomi, Illka, (2001) Internet, Innovation, and Oper Source: Actors in the Network, http://firstmonday.org/issues/issue6-1/-tuomi/index/html
Van de Ven, A., H., Delbecq, D., & Koening, R., Jr., (1976), Determinants of coordination modes within organizations, American Sociological Review, 41 (2), 322-338
Victor, B. and Boynton, A., C., (1998) Invented here: Maximizing your organization's internal growth and profitability, Boston: Harvard Business School Press
Wayner, P., (2001), Free for All: How Linux and the Free Software Movement Undercut the High-Tech Titans, New York, Harper Business
Yakhlef, Ali, We Have Always Been Virtual, (2002), book chapter in Organization and Work Beyond 2000, Eds. Jackson, P and Rapp, B. (forthcoming)
Zahra, S. & Covin, J., (1995), Contextual influences on the corporate entrepreneurship-performance relationship: a longitudinal analysis, Journal of Business Venturing, 10: 43-58
Zeleny, Milan, (1989), Knowledge as a new form of capital: Division and reintegration of Knowledge, Human Systems Management, 8 (1): 45-58