GNOME, a case of open source global software development

Daniel M. German
Department of Computer Science
University of Victoria
dmgerman@uvic.ca

Abstract

The GNOME Project is an open source project which main goal is to create a GUI desktop for Unix systems, and encompases close to two million lines of code. It is composed by a group of more than 500 different contributors, distributed across the world Some companies employ several of these contributors with the hope of accelerating the development of the project, but many other contributors are volunteers. The project is divided into several dozen modules, ranging from libraries (such as GUI, CORBA, XML, etc) to core applications (such as email client, graphical editor, word processor, spreadsheet, etc). This paper describes the organization and management of the project and describes the infrastructure needed by a contributor, how contributors work as independently together, but still with a common goal. It also describes how requirement gathering takes place, and its unique administration structure, rooted in the GNOME Foundation, a body created solely to oversee the current and future development of the project.

This is where the abstract go

1. Introduction

Bruce Perens describes open source as software that provides the following minimal rights to their users: 1) the right to make copies of the program, and distribute those copies; 2) the right to have access to the software's source code; and 3) the right to make improvements to the program [12]. Some of the classical examples of open source software are the Linux operating system, the Apache Jakarta project, or the GNU toolkit of software development tools (gcc, emacs, make). The GNU Network Object Model Environment (GNOME) Project (www.gnome.org) is an attempt to create a free (as defined by the General Public License, and therefore open source) desktop environment for Unix systems. It is composed of three main components: an easy-to-use GUI environment, a collection of tools, libraries, and components to develop this environment, and an "office suite" [6]. The GNOME Project was founded by Miguel de Icaza in 1996 as a loosely coupled group of developers, scattered all over the world. The project currently involves around five hundred developers. The first version (0.10) was posted in 1997. Version 1.0 was released in March 1999, a point in which it was integrated into Red Hat Linux as its default desktop. Version 2.0 is the latest stable version, released in 2002, and development of Version 2.2 is on its way. GNOME is composed of a large collection of programs and libraries, comprising almost two million lines of code [2, 3].

2. Infrastructure

One of the main requirements for distributed development is agreement on a common toolkit for software development, as each of the members of the team is expected to have access to these tools. In a private development, this is not an issue, as it is expected that the organization will provide the necessary software. Given the philosophy behind OSS and because volunteers are an important driving force in OSS, the toolkit of choice is usually OSS itself (one notable exception to this rule is the use of *bitkeeper* a commercial product for configuration management used by the Linux project, free to be used by Linux developers, with certain restrictions).

GNOME uses, like many OSS projects, the GNU toolkit (gcc, make, autoconf, automake, emacs, vi, etc), CVS for software configuration management, Bugzilla for bug management, GNU Mailman for its mailing lists, and of course, because its goal is to provide a desktop for Linux, it uses Unix as its development platform. Potential developers, by just installing a recent version of Linux, have an environment that allows them to start contributing to the project. The cost of entry is therefore minimized.

It is important to note that contributions to an OSS project are not only restricted to working code (patches, as they are commonly called); they can include documentation, bug reports, artwork, translations to other human languages, and many others. In this paper, we will use the term

developer to refer to any type of contributor to the project.

3. Project Management

In order to handle a project of this magnitude, the code base is divided into *modules*. There are four main groups of modules: a) required libraries (19 modules); b) core applications (4 modules); c) applications (16 modules), and d) other (several dozen modules and growing, these modules represent individual applications that are not considered part of the core of GNOME). Each module has one or more maintainers, who oversee the development of their corresponding module and coordinate and integrate the contributions of other developers to their module [4]. In a way, a module represents an independent product that can have its own maintainer, sets of requirements, development timeline. These modules are interrelated between themselves, but there relationships are kept to the minimum, so each module can evolve as independently as possible from the rest.

In an analysis of SourceForge projects, Krishnamurthy found that most OSS projects are composed of a handful of developers [9]. GNOME is one of the few projects that appears to break this rule, given that more than 500 people have "write access" to its CVS repository. Zawinsky, a Mozilla developer, provides insight to this phenomenon: "If you have a project that has five people who write 80% of the code, and a hundred people who have contributed bug fixes or a few hundred lines of code here and there, is that a '105-programmer project?" (as cited in [8]).

A preliminary analysis of the development logs appears to agree with Zawinsky's view. Most modules have few developers who write most of the code. For example, Evolution (the mail client of the GNOME project) is composed of approximately 160kLOCS (Feb. 2003) and almost 50% of the times the source code has been modified, the change can be attributed to one of 5 developers (see figure 1) [5].

The success of the GNOME project seems to lie in the division of the project into manageable modules in which a handful of developers (a team) can concentrate. The amount of communication between developers is therefore minimized.

As some modules grow into large entities themselves, they are then broken into smaller modules. For example, Evolution is broken into slightly more than 20 submodules, each as independent as possible from the rest. In includes modules dedicated to user interface, different mail libraries, filters, importers, documentation, translations, etc.

4. Requirements

As described in [13], most OSS projects do not have a traditional requirement's engineering phase. Specially at

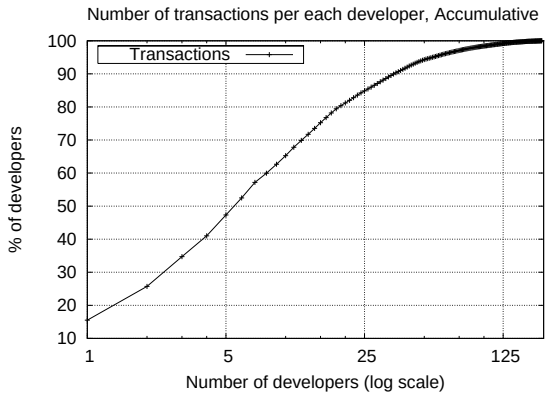

Figure 1. This plot shows the number of CVS transactions committed to Evolution per developer. From a total of 196 developers, 5 account for 47% of the CVS transactions, while 20 account for 81% of the transactions, and 55 have done 95% of them.

the beginning of GNOME, the only stakeholders were the developers who acted as users, investors, coders, testers, documenters, etc., with little interest in the commercial success of the project, but who, at the same time, wanted to achieve respect from their peers for their development abilities. One of the main goals of the developers is to produce software that is used by its associated community.

In particular, we can identify the following sources of requirements in GNOME:

- Vision. One or several leaders provide a list of requirements that the system should satisfy. In GNOME this is epitomized by the following non-functional requirement: "GNOME should be completely free software" (free as defined by the Free Software Foundation).

- Reference Applications. Many of its components are created with the goal of replacing similar applications. The GNOME components should have most if not the same functionality as these reference applications. For example, gnumeric uses Microsoft Excel as its reference, ggv uses gv and the kghostview, Evolution uses Microsoft Outlook and Lotus Notes.

- Asserted Requirements. In few cases, the requirements for a module or component are born from a discussion in a mailing list. In some cases, a requirement emerges from a discussion whose original intention was not to do requirement analysis. In other instances (as it is the case of Evolution), a person posts a clear question instigating discussion on the potential requirements that a tool should have. Evolution was born when sev-

eral hundred messages were created describing the requirements (functional and non-functional) that a *good* mailer should had before coding started.

- A prototype. Many projects start with an artifact as a way to clearly state some of the requirements needed in the final application. Frequently a developer proposes a feature, implements it, and presents it to the rest, who then decides on its value and chooses to accept the prototype or scrap the idea [7]. GNOME started with a prototype (version 0.1) created by Miguel de Icaza as the starting point of the project.

- *Post-hoc* requirements. In this case, a feature in the final project is added to a module because a developer wants that feature and he or she is willing to do most of the work, from requirements to implementation and testing. This feature might be unknown by the rest of the development team until the author provides them with a patch, and a request to add the feature to the module.

Regardless of the method used, requirements are usually gathered and prioritize by the leader of the project, the maintainer or maintainers of the module and potentially the Foundation (see section 6). A maintainer has the power to decide which requirements are to be implemented and in which order. The rest of the developers could provide input and apply pressure on the maintainers to shape their decisions (as in *post-hoc* requirements). Sometimes a subset of the developers group might not agree with the maintainer's view, and could potentially jeopardize the project, and create what is known as a fork (see section 5). So far this has not happened within GNOME.

5. Module Maintainers

Module maintainers serve the roles of leaders for their module. [10] identified the main roles of a leader in an OSS project as:

- to provide a vision;

- to divide the project into parts in which individuals can tackle independent tasks;

- to attract developers to the project;

- to keep the project together and prevent forking.

The success of an OSS project is dependent on the ability of its maintainer to divide it into small parts in which developers can work with minimal communication between each other and, with minimal impact to the work of others [10]. GNOME has been able to attract and maintain good, trustworthy maintainers in its most important modules due mainly by being employees paid to work on GNOME by different organizations.

5.1. The Paid Employees

As we described in [4], several companies have been subsidizing the development of GNOME. RedHat, Sun Microsystems, and Ximian are some of the companies who pay full time employees to work on GNOME. Paid employees are usually responsible for the following tasks: project design and coordination, testing, documentation, and bug fixing. These tasks are usually less attractive to volunteers. By taking care of them, the paid employees make sure that the development of GNOME continues at a steady pace. Some paid employees also take responsibility (as maintainers) for some of the critical parts of the project, such as gtk+ and CORBA (RedHat), the file manager Nautilus (Eazel, now bankrupt), the Evolution (Ximian), etc. Paid employees contribute not only in the form of code. One of the most visible contributions of Sun employees is the proposal of the GNOME Accessibility Framework, which aims at guaranteeing that GNOME can be used by a vast variety of users, including persons with disabilities.

In the case of evolution, the top 10 contributors (which account for almost 70% of the CVS commits) are all Ximian employees.

Volunteers still play a very important role in the project and their contributions are everywhere: as maintainers and contributors to modules, as bug hunters, as documenters, as beta testers, etc. In particular, there is one area of GNOME that remains done mainly by volunteers: internationalization. The translation of GNOME is done by small teams of volunteers (volunteers who usually speak the language in question and who are interesting in seeing support for their language in GNOME).

As with any other open source project, GNOME is a meritocracy, where people are valued by the quality (and quantity) of their contributions. We are currently evaluating the commit logs to analyze the amount and type of contributions from these paid employees to the project and how they compare to the contributions of volunteers. Most of the paid developers in GNOME were at some point volunteers. Their commitment to the project got them a job to continue to do what they did before as a hobby.

6. The GNOME Foundation

Until 2000, GNOME was a run by a "legislature" where each of its developers had a voice and a vote and the developer's mailing list was the "floor" where the issues were discussed. Miguel de Icaza served as the "constitutional monarch" and "supreme court" of the project, and had the

final say on any unsolvable disputes. This model did not scale well, and was complicated when Miguel de Icaza created Helixcode (now Ximian), a commercial venture aimed at continuing the development of GNOME, planning to generate income by selling services around it.

In August 2000 the GNOME Foundation was instituted. The mandate of the Foundation is "to further the goal of the GNOME Project: to create a computing platform for use by the general public that is completely free software." [1] The Foundation fulfills the following four roles [11]: 1) it provides a democratic process in which the entire GNOME development community can have a voice; 2) it is the responsible for communicating information about GNOME to the media and corporations; 3) it will guarantee that the decisions on the future of GNOME are done in an open and transparent way; 4) it is a legal entity that can accept donations and make purchases to benefit GNOME.

The Foundation is composed of four entities: its members (any contributor to the project can apply for membership); the board of directors (composed of 11 democratically elected contributors, with at most four with the same corporate affiliation), the advisory board (composed of companies and non-for-profit organizations), and the executive director. As defined by the Foundation's charter, the board of directors is the primary decision-making body of the GNOME Foundation. The members of the board are supposed to serve in a personal capacity and not as representatives of their employers. The Board meets regularly (usually every two weeks, via telephone call) to discuss the current issues and take decisions in behalf of the entire community. The minutes of each meeting are then published in the main GNOME mailing list.

6.1. Committees

Given the lack of a single organization driving the development according to its business goals, OSS projects tend to rely in volunteering to do most of the administrative tasks associated with that project. In GNOME, committees are created around tasks that the Foundation identifies as important. Developers then volunteer to be members of these committees.

Examples of committees are: "GUADEC" (responsible for the organization of the conference),, "Web team" (responsible for keeping the Web site up-to-date), "sysadmin" (responsible of system administration of the GNOME machines, the "release team" (responsible for planning and releasing the official GNOME releases), the "Foundation Membership" (responsible for maintaining the membership list of the foundation), and several others.

6.2. The Release Team

Each individual module has its own development timeline, and objectives. Planning and coordination of the overall project is done by the Release Team. They are responsible for developing, in coordination with the module mantainers, release schedules for the different modules, and the schedule of the overall project. They also keep track of the development of the project and its modules, making sure that everything stays within schedule. Jeff Waugh, a GNOME Foundation member, summarized the accomplishment of the team and the skills required (in his message of candidacy to the Board of Directors in 2002):

> *[The Release Team] has earned the trust of the GNOME developer community, madly handwaved the GNOME 2.0 project back on track, and brought strong co-operation and "the love" back to the project after a short hiatus. It has required an interesting combination of skills, from cheerleading and Maciej-style police brutality to subtle diplomacy and 'networking'.*

7. Communication

Developers are located in many different places all around the world. As described above, some are volunteers and the rest works for different organizations (and even within those organizations, they might still be located in different parts of the world, as it is the case with some Ximian developers). The GNOME communication relies on the following:

- Mailing Lists. The GNOME project has extensively used mailing lists. These lists have a wide range of purposes: some are intended for final users, some for particular components of the software, some for announcements, etc. Mailing lists provide a trail of decision making for the project.
- IRC: Internet Relay Chat. The "watercooler conversations" have been mimicked by using irc. Developers connect to a common IRC server/channel, and wait for other users to connect. The conversation is informal, with no real agenda.
- Web sites. The Web sites of the project comprise a large amount of information, intended for every type contributor to the project (coders, bug reporters, bug hunters, documenters, translators, etc).
- GUADEC, the GNOME Conference is a Foundation's effort to get developers together. Its goal is to provide a venue for discussion, interaction, and training. The

Foundation attempts to support many of the developers who cannot afford their own traveling expenses. This year's GUADEC will last 5 days and will take place in Dublin, Ireland.

- The GNOME Summaries. Every two weeks a summary is published in the GNOME mailing list. It usually contains the most relevant events of the period, links to new or improved documentation, news specific to different GNOME modules, a "Hacker Activity" section, enumerating the most active modules and the most active developers in the project, and a bug-hunting section, listing the number of current bugs per module including the progress made during the period.

8. Conclusions

After almost 6 years of development, GNOME has demonstrated to be a success. One of the major accomplishments of the project was the decision of Sun to replace its outdated CDE with GNOME. Its "free software" nature has created special requirements in the way that the project is organized and managed. Furthermore, GNOME is a project where people employed by different companies and volunteers work together with a common goal. Developers contribute in a wide range of ways (code, testing, bug reports, documentation, artwork, bug-hunting, system administration) and are located across the world, relying on the ability of its leaders and maintainers to manage the project, on the Internet as its communication channel and on several tools (such as mailing list, Web pages, CVS, Bugzilla) to maintain a good-enough communication that allows for the project to proceed. Recently, the Foundation has taken the responsibility of giving the project a coherent vision for the present and into the future, aimed at guaranteeing that GNOME continues to fulfill its main goal: "to create a computing platform for use by the general public that is completely free software."

About the author

Daniel M. German is a member of the GNOME Foundation and was the maintainer of ggv, the PostScript viewer for the GNOME project. He is currently assistant professor of computer science at the University of Victoria, in Canada. In his spare time he enjoys hacking free software.

References

[1] GNOME Foundation Charter Draft 0.61. http://foundation.gnome.org/charter.html, October 2000.

[2] J. Charles. Linux Support Ranges from GUI to Big Blue. *Computer*, 32(5):20–22, May 1999.

[3] M. de Icaza. GNOME History. http://primates.helixcode.com/~miguel/gnome-history.html, 2002.

[4] D. M. German. The evolution of the GNOME Project. In *Proceedings of the 2nd Workshop on Open Source Software Engineering*, 2002.

[5] D. M. German and A. Mockus. Automating the measurement of open source projects. Submitted for publication, 2003.

[6] T. Gwynne. GNOME FAQ. http://www.gnome.org/faqs/users-faq/, 2003.

[7] S. Hissam, C. B. Weinstock, D. Plakosh, and J. Asundi. Perspectives on open-source software. Technical Report CMU/SEI-2001-TR-019, Software Engineering Institute - Carnegie Mellon, November 2001.

[8] P. Jones. Brooks' law and open source: The more the merrier? does the open source development method defy the adage about cooks in the kitchen? IBM developerWorks, August 20, 2002.

[9] S. Krishnamurthy. Cave or Community? An Empirical Examination of 100 Mature Open Source Projects. *First Monday*, 7(6), June 2002.

[10] J. Lerner and J. Triole. The Simple Economics of Open Source.

[11] D. Mueth and H. Pennington. GNOME Foundation FAQ. http://foundation.gnome.org/faq.html, 2002.

[12] B. Perens. *Open Sources: Voices from the Open Source Revolution*, chapter The Open Source Definition. O'Reilly, 1999.

[13] W. Scacchi. Understanding the requirements for developing open source software systems. *IEE Software*, 149(1):24–39, February 2002.