

AN OPEN SOURCE SCREEN READER FOR BLIND AND VISUALLY IMPAIRED PEOPLE

Experiences and Thoughts

Martina Weicht, Frank Zenker, Ilvio Bruder and Andreas Heuer
IT Science Center Ruegen gGmbH, Circus 14, 18581 Putbus/Ruegen, Germany

Keywords: Open source software development, Community, eHealth, Screen reader.

Abstract: Developing open source software offers great potential for sustainable software evolution and for software product distribution, holding particular advantages in financial, psychological, security, and other aspects. However, lots of effort and fortunate circumstances are needed to enforce them. This paper will take a look at open source software development and its success factors using the example of the SUE screen reader project. This project deals with supporting blind and visually impaired users on Linux by providing alternative access to the graphical user interface. The development of SUE is an interesting example of FLOSS development in the eHealth area. We will introduce SUE and provide an overview of our open source development experiences discussing a special factor in the success of open source development: communities.

1 INTRODUCTION

Open source software development provides an interesting way for sustainable software evolution and for distributing software products. It offers advantages in various aspects such as in economy (efficient production of software based on available resources, better market position among competitors with similar, but non-free products, shorter time to market thanks to code reuse), in security (quick identification of potential security problems and their correction), and in personal psychology (peer recognition, engagement into a rewarding task) – just to name a few. For a more detailed discussion cf. (Working group on Libre Software, 2000). However, to enforce these aspects lots of effort and also fortunate circumstances are needed.

This paper will take a look at open source software development and its success factors using the example of the SUE screen reader project. This project deals with supporting blind and visually impaired users on Linux by providing alternative access to the graphical user interface. The development of SUE is an interesting example of FLOSS¹ development in general and in the health care area in particular.

In order to describe open source software develop-

ment in SUE along with its success factors and challenges, we will introduce the SUE project in section 2 and provide an overview of our open source development experiences in section 3. In section 4 we will discuss a special factor in the success of open source development: communities. Section 5 summarizes and discusses our experiences.

2 THE OPEN SOURCE SCREEN READER SUE

The project *SUE: Screenreader & Usability Extensions*² has been funded by the German Federal Ministry for Labor and Social Affairs for a period of 3 years (2007 – 2009). Its goal was to support blind and visually impaired computer users on Linux using the most important applications: desktop environment interaction, e-mail and text processing, spreadsheet calculation and a basic web browser support.

Development for SUE is based on the former Linux Screen Reader (LSR) project³. LSR was meant to be a “reusable development platform for building alternative and supplemental user interfaces in support of people with diverse disabilities” (Parente and

¹Free/Libre Open Source Software; free/libre/open in terms of source code access and reuse

²<http://sue.sourceforge.net/>

³<http://live.gnome.org/LSR>

Clippingdale, 2006). Its development was funded by IBM until a change in their accessibility support strategy (Parente, 2007b) led to the LSR project going dormant after one last release of version 0.5.3.

Looking carefully into the LSR project and all the work that had been done up to then, the decision was made to base screen reader development within SUE upon the LSR project – made possible by generous licensing of LSR.

For the SUE project, building upon LSR meant that we did not have to start from scratch. The LSR code base was carefully laid out as a sustainable, extensible architecture for assistive applications and allowed us to continue its development both along our ideas and a retrospect document provided by the LSR project lead (Parente, 2007a). Most of its architecture is still recognizable in SUE, especially the strategy of abstracting dependencies which was even improved in SUE. The code has been adapted to a more recent version of the used accessibility interface and restructured for better handling. Some of the actual screen reading functionality was only implemented in very basic ways in LSR, such as Braille support, which we integrated into a holistic output concept for all output capabilities in SUE: Braille, speech, and magnification. These are represented by both virtual and physical devices: a Braille display and a Braille monitor, magnification, and speech output (text-to-speech). Application support – both for the requested applications and a few small other ones – had to be updated and extended in some parts and newly created in others.

Today, SUE's base called *AccessEngine* is this envisioned platform for assistive applications and the screen reader SUE is the first of them; others are certainly possible. In general, the *AccessEngine* provides an event-based information processing to any input and output devices connected to it. It takes care of all data and device management tasks including receiving information from the accessibility interface (AT-SPI, part of Gnome Accessibility project⁴) and delivering it to the proper recipients.

Both the *AccessEngine* and SUE are published under BSD license⁵, so anyone may either support our work, develop their own open source screen reader based on SUE or develop another open source assistive application based on the *AccessEngine*. One idea may be to integrate it into an AAL scenario by turning the TV into an easy-to-use in-home information basis with speech output and sound icons.

⁴<http://projects.gnome.org/accessibility/>

⁵<http://www.opensource.org/licenses/bsd-license.php>

3 REQUIREMENTS FOR A SUCCESSFUL OPEN SOURCE PROJECT

There are quite a few information sources available on criterias for a successful open source project. Some of them provide rather practical guidelines such as “Use a version control system” or “Get free hosting for your project”⁶. Others describe more abstract success factors like pursuing a clear goal, maintaining focus on the project's mission, or choosing a business model⁷. With our background in software development, we have chosen four main success criteria and will apply them to the screen reader project.

3.1 Resources in Terms of Time, Knowledge, and Money

When the SUE project started, resources seemed plenty. The project was laid out for four full-time developers over a period of three years. None of them had any experience in either open source software development (though all have been using open source software before) or screen reader software, but had different levels of experience in software development in general (on Windows and Linux) and in project management. In order to gain insight into both of the new fields, a community of open source / accessibility / screen reader experts was brought together in a workshop.

3.2 A Clear Goal and a Goal-pursuing Strategy

After the workshop, the project team retreated to evaluating and interpreting this gathering, unifying its results with their project description while going into further research where necessary, into requirements analysis and project specification. This behavior has been criticized by the above-mentioned community as they did not feel involved into the decision-making process. They expected a more bazaar-like project organization (many contributors loosely organized via the Internet) whereas the project team – coming from closed source software development – adopted the cathedral style (code developed by small group and released to the public) (Raymond, 1999). This approach seemed natural to the project team as they were funded by the Ministry as opposed to the group

⁶<http://www.wikihow.com/Have-a-Successful-Open-Source-Project>

⁷<http://beradrian.wordpress.com/2009/05/12/success-in-open-source/>

of volunteers spending their free time on open source projects. Unfortunately, this misconception significantly harmed the parties' trust into one another, which for the remainder of the project did not recover.

Meanwhile with respect to their project description the development team made decisions for their development, defined a road map and milestones for their work and dove into screen reader development preparing first releases. When in doubt, the project description was used as a reference. At different stages during the project it differed from the community's opinion. However, it was decided to rely upon the official project description while at the same time respecting the community's interests and wishes as much as possible.

3.3 Marketing

For a long time SUE was too immature to be actively tested by volunteers. Disseminators were hesitant to approach possible users as testers as long as the screen reader only provided limited functionality – obviously it is difficult for a blind user to test a piece of software that he/she absolutely needs to rely upon. At the same time, others (namely the community mentioned in sections 3.1 and 3.2) kept asking for the source code to be published for them to take a look at and to be integrated into discussion and decision making – a contradiction between members of the community or even between two communities (a stable screen reader vs. early source code releases, screen reader users vs. open source community members / developers).

Our decision on publishing the source code presented a compromise to those two sides. Halfway through the project we chose to move our activities to SourceForge.net⁸, an online platform for open source developments. Up to then we have maintained a project wiki (in German), but did not offer our source code on the web. Two platforms have previously been considered on which to publish our development efforts. Gnome.org⁹ seemed to be a good choice, since for now SUE has been developed for the Gnome desktop environment as it provides the accessibility infrastructure that SUE relies upon. On the other hand, once SUE is no longer limited to Gnome desktops, this platform's label will be too restricting. We would mislead potential users or have to move again. Therefore, SourceForge.net was chosen as our platform of choice. Here it is a matter of adding a new keyword to the project's trove list when needed.

SourceForge.net proved to be a very useful platform for our project development as it provides a mul-

titude of tools for open source software development: web hosting, forums, blogs, mailing lists, trackers for bugs, feature requests, and patches, code control systems like SVN, Bazaar, Git, a file release system, project, content and task management tools, a wiki and more. SourceForge.net is well-established among OSS developers and only requires one registration for all activities within one or more projects. All information at SourceForge.net is provided in English in order not to limit ourselves to German contributions, but to find as many users as possible.

3.4 Sustainability

One of the biggest problems in open source development, especially with those projects that are funded for a certain period of time, is sustainability. After funding ends, development needs to continue in order for the software product to be used - even more with a screen reader that always depends upon an accessibility infrastructure on the one hand and on the applications to be supported on the other. As soon as development stops, the screen reader is out-of-date and next to useless.

In general, screen reader development cannot be done in projects. Other concepts and business models on how to permanently fund this work are necessary in order to ensure continuous development – even more as the implementation delivered after the funded project period is only a small part of the application support that is needed by screen reader users.

4 OPEN SOURCE COMMUNITIES – EXPERIENCES FROM THE SUE PROJECT

One of the crucial factors for open source software projects is their community of users, testers, bug reporters and fixers, and developers. This section takes a look at different types of communities, their establishment and difficulties related to open source communities from the viewpoint of the SUE project.

4.1 Autonomous and Sponsored Communities - How do They fit Together?

West and O'Mahony (West and O'Mahony, 2008) distinguish two different types of communities: autonomous (community-managed) and sponsored communities. As far as the SUE project is concerned,

⁸<http://sourceforge.net/>

⁹<http://www.gnome.org/>

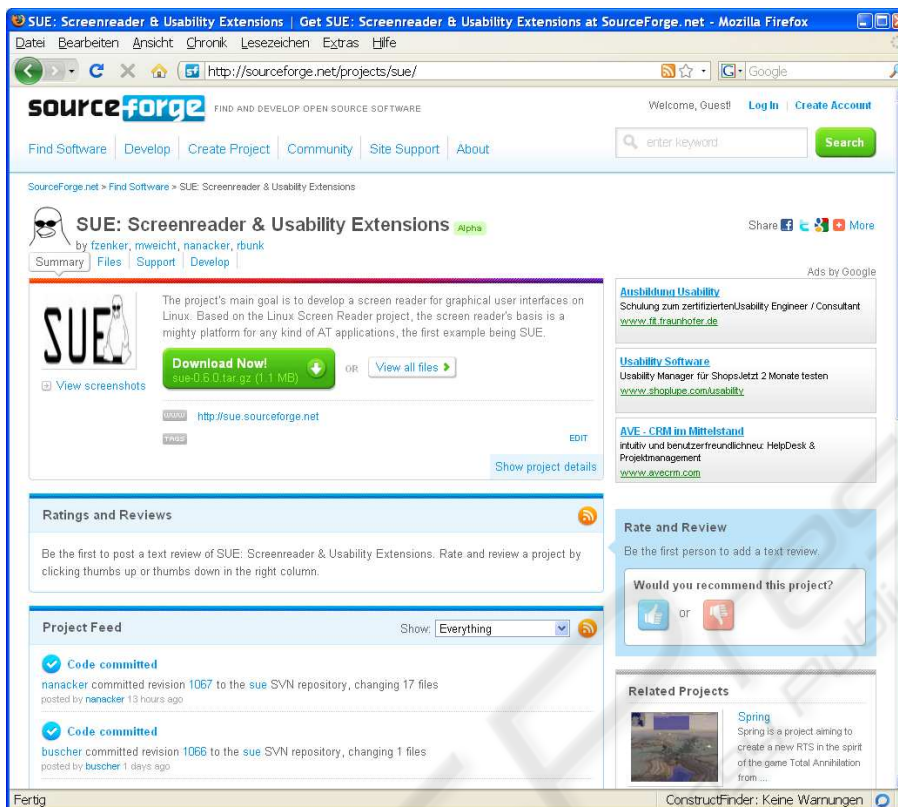


Figure 1: The SUE screen reader project at SourceForge.net.

the project team can be seen as a sponsored community with funding provided for a certain period of time.

On the other side, there is an autonomous community of people interested in Linux accessibility. In this particular setting, this community was the one holding the information needed by the SUE project team during their early project phase (cf. section 3.1). Serious mistakes in communication have been made in this phase; to a certain degree due to how the project team interpreted their own role: the cathedral – “a small group of core developers who control the architecture and direction of development” (Scacchi, 2007). The group decided to work on the screen reader basis and main concepts first before putting it up on the web for everyone to look at and start contributing. Speaking in terms of West and O’Mahony and their two types of openness in communities, the project team decided against accessibility, meaning the direct influence of external participants on the project’s direction, while lacking transparency in the sense of letting the community know what was happening in the project. As described in section 3.2, this policy of non-communication caused irritation within that community and led to a few counterproductive reactions on their side. At this stage, both communities

did not regard each other as partners.

This situation influenced the project’s standing when roles were about to change. By moving all development efforts to SourceForge.net (cf. section 3.3), the team offered more transparency in terms of project activities. The software gradually became more and more elaborated and the need for testers grew. The community that was supposed to be the screen reader’s target group still suffered from what they perceived as ill-treatment in the first phase of the project, ignored the development work or eyed – and mouthed – it very critically.

Questions arose on why they should spend their time on a project that others were paid to work on.

4.2 From a Sponsored to an Autonomous Community

Despite this unfortunate start, what is desirable for this particular project – and any project receiving only initial funding – is to be transformed into an autonomous community project. In order for this to work, volunteers need to be acquired and encouraged to go through the various stages of community membership. Nakakoij et al. (Nakakoji et al., 2002) mention eight different roles of community membership

(passive user, reader, bug reporter, bug fixer, peripheral developer, active developer, core member, and project leader). These may not all be present in each and every open source software project, but the general classification holds true. Those categories, however, are not static. As opposed to traditional software development, community members may change their roles as they gain more insights into the software and acquire more skills. Nakakoij et al. identify two factors for a successful evolution of an open source software community: “the existence of motivated members who aspire to play roles with larger influence, and the social mechanism of the community that encourages and enables those individual role changes.” What motivates people to get engaged in an open source project? Numerous reasons exist. Scacchi names “self-determination, peer recognition, project affiliation or identification and self-promotion, as well as the belief in the inherent value of free software.”











This did not hold true for the community of Linux accessibility enthusiasts mentioned before as they did not find a way to overcome their prejudices nor did they identify with the project.

At this point it was clear to the development team that their community of interested users needed to be found somewhere else. The community of visually impaired people, however, is not a large one. Their fraction of people interested in Linux is even smaller. Apart from this small number of potential new community members, what are the difficulties people face when considering joining as open source software project? Shibuya and Tamai (Shibuya and Tamai, 2009) identified a number of obstacles such as selecting “a suitable task according to his/her skills, lack of up-to-date development documents, constraints imposed by contributor licensing agreement, no response from core developers for their doubts and support requests, and need to learn a new tool”.

This clearly shows responsibilities both on the newcomer’s and the community’s side that need to be taken care of in order to successfully integrate the newcomer into the open source project community.

Something that worked well already in the tiny SUE community was localization. As the project is still very young, we have decided not (yet) to rely on global translation projects such as the Gnome localization team. Again, this would have meant a close coupling to Gnome, plus we needed in-time localization and flexibility for our work.

Also we were hoping to keep translators beyond that task, engaging them in testing and/or development as well. SourceForge.net provides a “Help

 English (US)	100 %	intrasystem language
 English (GB)	100 %	for testing purposes
 German	100 %	project-immanent
 Hungarian	97 %	business contact / student
 Romanian	88 %	project partner / staff member
 Dutch	69 %	contact over SourceForge.net
 Swedish	60 %	contact over SourceForge.net
 French	54 %	private contact
 Portuguese	47 %	private contact
 Czech	12 %	private contact

Last updated: 16-10-2009

Figure 2: The status of internationalization for the SUE screen reader.

wanted” section as a nice feature to actively ask people to participate. A little note in that section asking for help in localizing SUE lead to four inquiries by interested users, two of whom actively translated it into their mother tongue. Translating SUE is a clearly defined task with manageable effort (about 1.100 words and strings). Little expertise is needed, i.e. no special programming skills are required. Neither of them was a screen reader user or had any previous experience with assistive software. Both of them were involved in the project on a very low level and even though translations need to be updated as development progresses, to them their contribution did not mean a long-term commitment to the project.

4.3 What are the Difficulties in Establishing an Autonomous Community?

With a small target group like ours, establishing a community takes time. Users will not pour in by themselves as in our case people are used to and dependent upon their (Windows) screen readers. Moving over to Linux is a big step for a lot of them, one they do not necessarily perform by themselves. Instead, it needs external forces such as their workplace moving to Linux and therefore creating the need to learn Linux and a screen reader on Linux. If this does not happen, lots of marketing effort is needed such as presenting the screen reader at assistive technology exhibitions and conferences and directly approaching disseminators like our project partner, the Study Centre for Visually Impaired Students (SZS) Karlsruhe¹⁰, has done with their teaching concept on SUE screen reader and Linux usage. By providing hands-on introductory sessions, visually impaired people got to test both the screen reader and the Linux OS, namely

¹⁰<http://www.szs.uni-karlsruhe.de>

Ubuntu, in order to get an idea of what using Linux with a screen reader feels like.

In return, the development team received valuable feedback on their work from participants, who over the better part of the second project phase were the only testers of the screen reader. Unfortunately, only very few kept using the screen reader. This is partially due to the early development stages the screen reader was in during their teaching sessions.

Still, little by little, people are getting interested in the project like one person who looked into screen reading software for his blind friend. He used the forum provided at SourceForge.net to ask questions about the screen reader and interaction with him has been very pleasant and constructive.

On the other hand a few issues arose during project work that need to be considered as well.

Competing Projects. When starting the project, another Linux screen reader, Orca¹¹, existed. This project is funded by SUN Microsystems, Inc. and relies upon a large community of users, testers, bug reporters and fixers, and developers. While competing products are a healthy mechanism in economy, in open source software development they are often regarded as waste of time and development powers. “Instead of creating a second cripple, why don’t you put your efforts into helping that first cripple improving?” (Comment given by one of the community members).

Resource Management within the Core Development Group. While trying to establish a community, there is a constant weighing up of the actual development tasks against the efforts of communicating with potential community members. Even though talking about the software and explaining it to outsiders might provide new insights to the developers themselves, it is nonetheless time-consuming and requires patience. If available, a broker might help in this situation (cf. section 5).

5 DISCUSSION

A number of obstacles have complicated project development within SUE. The various communication problems might have been preventable, had the project team known more about how open source software projects work and had roles been clearly defined. More transparency along with a strategy on how to integrate the community of Linux accessibility enthusi-

¹¹<http://live.gnome.org/Orca>

asts into the project and at which stages of the project might have prevented some of these issues.

It would be interesting to see if a broker mediating between the two groups as described by Wenger (Wenger, 1998) could improve communication and how such a broker gets to be accepted by both groups. Finding the right broker would be crucial though as much of the communication success depends on his/her skills. Should the broker be sent by one community? Should he/she be completely independent from both groups? Involving this extra person requires even more sophisticated communication structures as this person needs to know exactly what both groups do and what they want to do in order to provide a satisfying assistance to both.

Without a broker, all project members need to be willing to interact with their community (of which they are a part) and the time needed to do so must be included in their work packages. This procedure is even more important with the usually different area of expertise within the development team.

6 CONCLUDING REMARKS

Without FLOSS, the SUE project would not have been as successful as it now is. Being able to build upon the LSR project saved a lot of time and allowed to progress faster than with an implementation from scratch. SUE also uses existing open source projects to extend its functionality, i.e. Braille support and screen magnification.

Its success, however, does not only depend on its source code and the functionality it provides, but on its community of users, testers, and developers. In order for it to grow, source code accessibility and project development transparency are of great importance. They require rethinking from non-open source projects where development takes place within the development team and a final product is released.

However, with sufficient knowledge on the mechanisms of open source software development in communities, engaging into one can be a very rewarding task with large potential for the software and a great personal experience.

ACKNOWLEDGEMENTS

The SUE project team would like to thank the German Federal Ministry for Labour and Social Affairs for funding this project and expresses their gratitude towards everyone who helped SUE along the way.

Thank you very much for your constructive feedback, help, and support!

REFERENCES

- Nakakoji, K., Yamamoto, Y., Nishinaka, Y., Kishida, K., and Ye, Y. (2002). Evolution patterns of open-source software systems and communities. In *IWPSE '02: Proceedings of the International Workshop on Principles of Software Evolution*, pages 76–85, Orlando, Florida. ACM.
- Parente, P. (2007a). Linux screen reader in retrospect. <http://www.gnome.org/parente/lsr/retro/>.
- Parente, P. (2007b). Status of IBM accessibility. <http://mail.gnome.org/archives/gnome-accessibility-list/2007-June/msg00000.html>.
- Parente, P. and Clippingdale, B. (2006). Linux screen reader: Extensible assistive technology. In Keates, S. and Harper, S., editors, *ASSETS*, pages 261–262. ACM.
- Raymond, E. S. (1999). *The Cathedral and the Bazaar*. O'Reilly & Associates, Inc., Sebastopol, CA, USA.
- Scacchi, W. (2007). Free/open source software development. In *ESEC-FSE '07: Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, pages 459–468, Dubrovnik, Croatia. ACM.
- Shibuya, B. and Tamai, T. (2009). Understanding the process of participating in open source communities. *Emerging Trends in FLOSS Research and Development, International Workshop on*, 0:1–6.
- Wenger, E. (1998). *Communities of practice - Learning, meaning, and identity*. Cambridge University Press.
- West, J. and O'Mahony, S. (2008). The role of participation architecture in growing sponsored open source communities. *Industry & Innovation*, 15(2):145–168.
- Working group on Libre Software (2000). Free software / open source: Information society opportunities for europe? version 1.2.

