

Open Source As a Knowledge Management Instrument

Thomas Wieland

Department of Electrical Engineering and Computer Science
University of Applied Sciences Coburg
Friedrich-Streib-Straße 2
96450 Coburg
Germany
thomas.wieland@fh-coburg.de

Abstract: Knowledge management typically focuses on communicating and spreading information and knowledge within a company. It is usually limited to technology or business information in spoken or written form. The representatives often overlook, however, that the code of software programs is also a perfect medium for denoting and communicating knowledge. In this paper, we want to show the benefits and problems that are related to a free exchange of code within a corporation. Applying experiences and best practices of the public open source software community to corporate communities leads to several preconditions that are essential for the success of this approach. Even a substantial change of the company's culture may be connected with the introduction of corporate open source.

1 Introduction

The most important asset of companies in our age is the knowledge they have gathered. Much of it is documented on paper, as patents, best practices reports, business process guidelines, or projects files. Even more knowledge resides in the heads of the employees. This is the reason why many companies have established numerous knowledge exchange instruments, summarized by the notion of “knowledge management” (KM) [Al95], [Ma02], [Ro00].

Even a coffee machine may be such an instrument. People meet there more or less accidentally and begin to talk about each other's experiences and problems. A more formal way are Intranet portals. They may list the experts of the company and allow searching for expertise about a particular topic. A KM portal may also be designed to cover all information about a certain subject, e.g. the products of competitors and their strengths and weaknesses or a technology like Java programming. Other KM instruments are internal workshops and conferences where the employees can pass on their knowledge to others, exchange their views, and discuss strategies.

The over-all goal of all knowledge management activities is to gain, store and provide knowledge for improvements in business processes and product development. The communicative instruments of knowledge management may be applied to software development, but cannot really cover the highly effective approaches for code re-use.

The management often does not realize that the lines of code of a software product can be pure knowledge. They have been elaborated by engineering art, tested, reviewed, and refined. As any knowledge, the value of a particular piece of code may vary to a large extent, from wrappers and work-arounds to complex algorithms and clever architectures. So the amount of knowledge store in the repository “code” can certainly differ very much.

It is obvious to most companies that software is an intellectual property that needs to be protected from theft and misuse. But the consequences of this external view to the internal handling are often missing. As soon as software code is accepted as valuable knowledge, it becomes clear that it is an issue for corporate knowledge management. As the usual KM instruments more or less fail for software, other means of exchange and communication have to be developed. For this purpose, the experiences and processes of the open source community can be extremely helpful.

In this paper we want to describe some of the most relevant organizational and technical questions that arise when beginning to set up a corporate software exchange as well as some of the major consequences.

2 Corporate Software Communities

Open source is a well-known paradigm for the royalty-free exchange of software in source code. (Note: It is sometime also called “free software”. In this paper we will only use the notion of “open source software” (OSS), meaning software whose distribution license meets the criteria of the Open Source Initiative [OSI04].) The original ideas were that software industry deprived the user of its natural rights to adapt software to his needs. When software is free – in the sense of “free access” – the user has the right to copy it, distribute it, analyse it, and improve it. The decisive precondition for realizing these rights is that the source code of the software is available.

2.1 Open Source in Commercial Contexts

Meanwhile the open source model has gained significant importance. The most prominent example is the free operating system Linux. Although exact figures are hard to derive because of its free character, Linux has already a considerable share of the server market. Other open source programs like the web server Apache, the GNU programs with compilers, configuration tools etc., the data bases MySQL and Firebird, or the office suite OpenOffice are equally successful.

All are distributed under an open source license, are available free of charge and can be used, modified, or improved by anybody. Many companies have realized the potential of the open source model and are successfully making use of various business models [He99].

In the course of this development, the organization of many projects has become more professional. Practices from development in enterprises have been applied to OSS projects and – vice versa – open source ideas have found their way into programming departments [Ha04]. As soon as a commercial development team knows how to integrate open source software into its development cycle, it makes no difference anymore where this OSS comes from. From entrepreneurial and legal points of view it may be, however, crucial if the software comes from outside or inside of the company.

Using and adopting a piece of OSS for a commercial development project is a more or less well-known practice (although some pitfalls still exist). The problems and discussions arise when formerly proprietary software should be released as open source. Large companies, called “corporations” in the following, with a broad product spectrum and several separate development divisions can benefit in the first way without risking the second way.

2.2 Advantages of Corporate Open Source

The solution is to set up an open source culture within the boundaries of the corporation. The number of potential participants is certainly much smaller than in public communities, but already with a couple of unconnected teams a vivid exchange can be established. The application of open source concepts to a corporate environment brings several advantages (cf. [DG01]): Software development needs not start from scratch every time as it is still done frequently, but can reuse a pool of existing code. So the expenses for the individual projects are reduced leading finally to lower costs and shorter time to market. Moreover, errors can be detected and investigated by independent users. Since the corporate reputation of the authors may be at stake, the quality of the code can be higher. Finally, the switch from one project to another is smoothened if the same basic code pieces and coding conventions are used everywhere.

3 Necessary Preconditions

In order to exploit the full potential of corporate open source communities, several preconditions must be fulfilled. These cover the entire software development process of the company and go through all levels. They can certainly realized step by step only, but all departments should be aware that the success of free internal software depends essentially on these preconditions.

For the migration we recommend a two dimensional procedure. On the one hand, a small number of organizational units starts with setting up the preconditions and releasing some of their own code. Later this process can be extended to more and more departments. On the other hand, a precisely defined technological area (e.g. CORBA, GUI components, etc.) should be selected in which the first software is offered. It must be assured that all development activities in this area follow common guidelines and conventions. In the course of time, more and more areas may be included.

The following preconditions are at least necessary for an effective exchange of free software:

- For using and improving program code in multiple environments it is required that other developers can easily comprehend the code. So common conventions for comments, naming of identifiers, indentation etc. are highly recommended.
- The exchange of software over project boundaries can only reach a critical mass if the number of operating system platforms that are used company-wide is limited and their configurations are clearly defined and maintained.
- The reuse of code is leveraged if any dependencies to commercial libraries are avoided. Thus the usage of such libraries is generally not recommended. The tools used in the development process (e.g. compiler, configuration management, version control) should also be freely available to any corporate programmer.

But the success of the migration eventually depends more on how efficient the culture of the company can embrace the virtues of the open source community. The most important aspects can be shortly mentioned here only:

- In conflict situations the common wealth of the entire company must be allowed to be put above the wealth of the individual group or department.
- Project leaders must convince by competence and not only by appointment due to hierarchy.
- Developers with the required skills must be available in a sufficient number. They must have the freedom to expand their knowledge to new, upcoming technologies, even if these are not currently in the focus of the department.

The implementation of these organizational preconditions must be accompanied by the fulfilment of some technical preconditions as well, e.g. the establishment of a central code repository, informal electronic communication means like mailing-lists and portals, an authorization and access model that allows to track down the origin of any repository entry, or flexible search mechanisms.

4 Conclusion

The company-wide exchange of program code and common development across organizational structures in an open source style can generate many synergies. It can ultimately reduce unnecessary double expenses and lower development costs. As program code is an excellent formal container for knowledge, it is undoubtedly an issue for any knowledge management representative to encourage the free exchange of code. Some of the central preconditions for a successful set-up of a corporate community have been sketched above. Several others had to be omitted, e.g. all legal questions. They depend to a large degree on the legislation under which the company operates. Therefore general recommendations are not possible.

The most important task in the introduction process of corporate open source is, however, the enforcement of the necessary organizational changes. This can eventually mean massive changes of the corporate culture which may be painful and require much convincing. But there is hardly any alternative: open exchange of information and efficient use of all information assets will soon belong to the core competences of all software-related businesses.

Acknowledgements

This work was in parts supported by Siemens AG and by the German Federal Ministry for Education and Research (project “NOW”). The author wants to thank all partners of the project for fruitful discussions and cooperation.

References

- [Al95] Alvesson, M.: Management of knowledge intensive companies. Berlin, 1995.
- [DG01] Dinkelacker, J.; Garg, P. K.: Corporate Source: applying open source concepts to a corporate environment. In: Feller, J.; Fitzgerald, B.; van der Hoek, A. (Eds.): Making Sense of the Bazaar: Proceedings of the 1st Workshop on Open Source Software Engineering, Toronto, 2001.
- [Ha04] Hang, J.; Hohensohn, H.; Mayr, K.; Wieland, T.: Benefits and Pitfalls of Open Source in Commercial Contexts. In: Koch, S. (Ed.): Free/Open Source Software Development. IDEA Group, 2004.
- [HO01] Hars, A.; Ou, S.: Working for free? - Motivations for participating in open source projects. In: Proceedings 34th HICSS Conference, Maui, 2001.
- [He99] Hecker, F.: Setting up Shop: The business of Open Source Software, www.hecker.org/writings/setting-up-shop.html, 1999.
- [HNH03] Hertel, G.; Niedner, S.; Herrmann, S.: Motivation of software developers in open source projects: An internet-based survey of contributors to the Linux kernel. Research Policy, Special Issue, 2003.
- [Ma02] Maier, R.: Knowledge management systems. Berlin/Heidelberg, 2002.
- [OSI04] Open Source Initiative: The Open Source Definition (Version 1.9). <http://www.opensource.org/docs/definition.php>, 2004.
- [Ro00] Roy, R.: Industrial knowledge management. London, 2000.