

Career in Open Source? Relevant Competencies for Successful Open Source Developers

Karriere in Open Source? Relevante Kompetenzen für erfolgreiche Open Source Entwickler

Nicole Kimmelman*, Friedrich-Alexander-University Erlangen-Nürnberg

* Correspondence author: nicole.kimmelman@wiso.uni-erlangen.de

Summary Open Source (OS) offers new ways of career for software developers. The article describes relevant competencies in a systematic structure along characteristic principles and challenges in Open Source projects. The results are based on a Grounded Theory content analysis of interviews with Open Source software developers, their project managers and human resource managers in Open Source software companies. Implications for future Human Resource Management in software companies are presented as an outlook. ►►► **Zusammenfassung** Open Source bietet für Software-Entwickler neue

Wege der Karriere. Der Beitrag skizziert die hierfür notwendigen Kompetenzen in einer systematischen Struktur entlang von charakteristischen Merkmalen und Herausforderungen der Entwicklung in Open Source Projekten. Die Ergebnisse basieren auf Interviews mit Open Source Software Entwicklern, Entwicklungsleitern und Personalentwicklern in Open Source Unternehmen, die im Sinne eines Grounded Theory Ansatzes ausgewertet wurden. Handlungsempfehlungen für zukünftige Personalentwicklungsmaßnahmen in Software Unternehmen bilden den abschließenden Ausblick.

Keywords ACM CCS → Software and its engineering → Software creation and management → Collaboration in software development → Open source model; ACM CCS → Social and professional topics → Professional topics → Computing education
►►► **Schlagwörter** Open Source, Software-Entwickler, Kompetenzen, Karriere

1 Introduction

Former analysis of OS software communities has addressed status dynamics within OS communities [32], contribution patterns [21] and how they create informal and formal social structures to manage membership and joining processes [38]. But there is a lack of knowledge to understand which competencies are necessary to reach specific status in OS projects. As a result ways of OS career seem to be quite incalculable for developers so far.

Which competencies are relevant in order to work as an OS developer in comparison to a developer job in the proprietary software sector? How should and could these competencies be supported by human resource management strategies in OS software companies?

The article tries to satisfy the need for empirical data to answer these questions from a Human Resource perspective. Relevant competencies will be summarized systematically in a table. The needs and strategies of a holistic Human Resource Management in software companies to promote the developers in their competency profile are presented structured in a model at the end of the article.

The answers are of interest to software developers deciding whether contributing to OS. But the results should also be relevant for OS/Closed Source software companies as far as the status and success of their developers in OS projects has positive economic effects on their own product planning and distribution. Human resource

management for software developers does not address these new challenges in a systematic way so far (not as reflected in the literature). The implications at the end of the article might be an interesting starting point to think about the career development of OS software developers in a more comprehensive way.

2 Open Source as a Field of Career for Software Developers

2.1 Career Objectives of Developers Participating in OS Projects

Since the increasing importance of OS software within the late 1990s there has been a lot of research from different disciplines on the motivation of software developers to contribute to OS [1; 3; 9–11; 16]. Especially the apparent altruistic character of developers as demonstrated by freely contributing to open software projects has been studied by researchers [11; 13]. Two main ranges of explanations can be defined [4]. Following the anthropological line of argumentation, OS engagement of participants is determined by the motive of improving the software to their own needs [e.g., 10] and their willingness to invest time because of fun [15; 35]. The idea of “intrinsic motivation” [8] is connected to this ideal of a gift based OS community linked to reciprocity and kinship.

More economically oriented studies showed that individual rationality of the developers in order to maximize “profits” such as improving their own programming skills [e.g., 11; 16; 36], or earning reputation and credits [6; 15; 27; 30; 39] must be seen as equivalent inducement for an OS engagement. For example: at least 61% of respondents in the Bitzer et al. investigation [3] believed that OS activities benefit their career.

Consistent with these results scientific literature has noticed a growing phenomenon of developers creating OS software as a component of their paid employment [1; 23; 26]. This includes strategic integration of OS in the developer's own professional career profile. Developers who perform in OS projects successfully might be able to offset their opportunity costs by signaling marketable skills to future employers. This can lead to higher job versatility [19] and higher wages [18].

Despite of those theoretical deliberations towards the (monetary) effects of OS activities there is hardly any empirical research about the real careers of OS developers. Existing studies only show that OS engagement by itself does not guarantee better career chances. Actually it is the committer status earned in those OS projects that really correlates with higher wages – at least when it comes to merit-based Foundation projects [9]. That emphasizes the importance of a particular community career path which developers need to follow in order to achieve positive effects from their OS involvement.

2.2 Relevant Principles of Career in OS Projects

Unlike other occupational communities OS communities and their developers are not necessarily associated with

a single employer or workplace. Instead members of the community are likely to work toward collective goals outside of their employment [27]. Rules and routines within the OS community are established by itself. Therefore it is interesting to have a closer look at characteristic principles of career in OS communities.

OS is ideal-typically characterized by an egalitarian process of collaboration/contribution, a public discussion tradition and a merit-based decision making process [14; 17; 22] in order to attract high-quality contributions from voluntary members [25]. Égalité does not mean that every developer contribute in the same way and amount. But people working in the OS Community find their own processes and best projects in a self-organizing way [29]. This openness allows “the results of creativity to be used, developed, and tested by anyone so that everyone can learn from one another” [12, p. 140]. Developers are able to maintain different roles in this process and to develop their own competencies quite freely in contrast to other disciplines. But the community “must integrate the individual contributions into a common pool, which can heighten interdependencies and the need for coordination mechanisms (e.g. Thompson, 1967)” [25, p. 1081]. Successful OS projects often have strong leaders [21; 22].

These distinctive principles influence careers in OS projects as well: In contrast to closed software development, positions in the OS Community are not only assigned but need to be earned in the opinion of the community. The community members have a big influence in promoting someone implicitly or explicitly. That means successful developers must be willing and able to fulfil the philosophy and criteria of the community. Merit will be rewarded with greater status, responsibility, or opportunity to enhance the development of the OS software [32; 38]. The following model of a typical career path within OS projects illustrates these principles on a more specific level. It is based on the three main positions identified in different OS projects: user, contributor and committer. Of course, the presented model can only be interpreted in its simplification as OS projects differ in their complexity, hierarchy and forms of interaction between their members [25]. The outlined career path needs to be seen in its exemplarity – integrating typical roles and activities confirmed in different more complex models [e.g. 20]

As Fig. 1 shows, reaching the highest level called “committer” can only be achieved by a multiple-stage process from user to contributor up to committer. This career requires first of all being an active part of the OS project community. This communal process reduces the risk of a wrong decision [29].

The first career position as a contributor seems to be achievable quite simply for gifted and encouraged developers as there are no entry barriers for users to the OS market. Keeping in mind the never ending need for new features to improve the quality of OS software developers can select their favourite project to work on their expe-

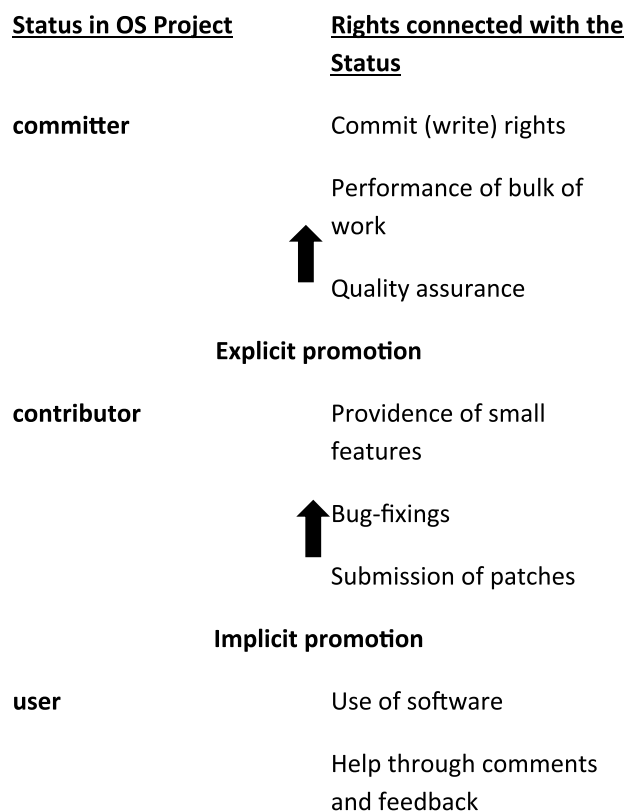


Figure 1 A typical career path in OS projects. **Origin:** Own figure based on Riehle [29].

riences and reputation. Contributors differ from normal users insofar as they are recognized and accepted by the community. The contributor status enables to submit patches, fix bugs or provide small features. Besides, the status does not have a sustainable advantage but takes a lot of time and effort.

The second career step to the level of committer is connected with an explicit promotion by the existing committers. That authority is not handed out easily. In other words: Entrance for new developers into OS projects on this level is restricted as the achieved status is usually permanent. This limits the possible number of committers in one project.

In contrast to the first career step more than just technical competencies seem to be relevant to get this grant. Leading positions require skills in building the organization [25]. Finding out more about these necessary competencies was the main purposes of our research project at the University of Erlangen-Nuremberg.

3 Methodology

3.1 Research Design

The results are based on a grounded theory research design [5] using guided interviews with two OS software developers (Interview 1 and 2), their project managers (Interview 3 and 4) and human resource managers (Interview 5 and 6). Each interview was approximately

90 minutes long, following a particular manual for the different sets of interviewees. In order to get access to the field but also to combine the Informatics and Human Resource perspectives, the interviews were led by a team consisting of a professor for OS software (Prof. Dr. Dirk Riehle, University of Erlangen-Nuremberg) and an assistant professor for professional competency development (Prof. Dr. Nicole Kimmelmann, University of Erlangen-Nuremberg). The theory building process was characterized by triangulation of data collection and data analysis which made it possible to integrate interesting aspects of the first set of data into the sampling and data collection of the second company/group of participants. Participants were asked to describe relevant competencies of OS software developers depending on their stage of career/position in OS projects. Possibly connections between the status in the OS community and the position/status in their OS company were particular interesting as well. Relevant competencies for the future were discussed at the end of the interviewing process in order to deliver implications for a Human Resource development programme within software companies. The three perspective format (including the Human Resource managers) made it possible to gather data from the practitioners' point of view but also to be aware of their "blank spots" when it comes to their own relevant competencies.

3.2 Sampling

The interviewees were chosen following the idea of theoretical sampling [5] from two companies developing OS Software based on the Linux kernel. Both companies are well established in the OS Software field with their own OS products. The actual selection of the companies was also based on the idea of contrasting a small and a big OS software company. The number of employees ranges from 45 to 350 people. All interviewees were selected by the companies itself and took part freely in the research project. Both integrated software developers had at least 10 years working experience in the field of OS development and were committer in several OS projects. The project managers have been developing software in OS Software companies/projects for more than 15 years. Their teams are involving 13 respectively 45 software developers at the moment. The Human Resource managers have both been part of the companies for more than five years and were responsible for professional development in similar positions in companies outside of the field of informatics.

3.3 Quality of the Results

All interviews were transcribed and analysed using qualitative content analysis software called MAXQDA for a systematic and transparent process. Similar aspects and corresponding concepts were categorized in a multidimensional hierarchy of competencies. The analysis of the interviews was discussed by both researchers in a pair analysing process in order to increase the internal validity

of the data analysis. The presented results are to be seen as confirmed in both samples. Conflicting points shown in the interviews must be analysed in further research steps including a broader sampling and range of interviews.

4 Relevant Competencies of Successful OS Software Developers

The following table summarizes the necessary profile of successful developers on their way to the committer status as it was recorded in the interviews. The structure of the table connects distinctive marks of the OS Software community and work with relevant competencies of the developers in order to systematize the competencies. Referring to the often cited educational competence model of Erpenbeck and Von Rosenstiel [7] the described competency profile distinguishes between technical (T), social (S) and personal (P) competencies. Technical competencies are relevant technical knowledge (like programming in Linux), documented technical experience and corresponding attitudes that are relevant for the successful implementation of OS software. Social competencies are understood as interpersonal skills required to support the software development and distribution process or the person's own career in an explicit way. Personal competencies include attitudes, values and motivation of the software developer. Besides, they are strategies and skills for organizing one's personal life and to developing their own personality. These factors can support or hinder a successful career of the developer by regulating his/her professional behaviour. They are usually quite stable and strongly connected to the personality of the developer. Nevertheless they can be changed by an inner process of self-reflection [7].

In the following discussion important aspects of the results are highlighted and explained further.

- **High Importance of Social Competencies**
Being a successful member of an OS project is not limited to technical competencies. The interviewees emphasized a "well-balanced competency-profile" (Interview 1) for successful developers. Indeed it is the social competency field that is crucial across all levels of career (Interviews 1–6). The high importance of these competences rejects the prejudice of the social incompetent "nerd" in a very comprehensive way.
- **High importance of communication skills and team work**
Communication skills and ability for team work are the most important competencies at all as developing OS software is mainly organized by email-communication and through global virtual teams (Interviews 5 and 6). Relevant competencies regarding this kind of communication are summarized in the category "e-Mail-competency". This includes skills like answering questions of other users friendly (Interviews 1 and 4) and with respect to their (cultural) communication style (Interview 2) or dealing with

Table 1 Relevant Competencies of Successful OS Software Developers.
Origin: Own table based on empirical data.

Distinctive characteristics of OS software Development	Relevant competencies of OS developers
Egalitarian process of collaboration/contribution	Technical: • Programming • "Architecture competency" • "Implementation of new features without disturbing others' work" • Quick induction into new projects Social: • "E-mail-competency" • Capacity for teamwork • "Not being arrogant against others" Personal: • Altruistic character • "To want to be in on the whole" • Motivation to improve software • Motivation through acknowledgement
Philosophy of "social give and take"	
Tradition of public discussion	Technical: • Implementation of feedback Social: • Target-group-specific communication skills • Giving constructive feedback • Compliance with social rules Personal: • Ability to take criticism • "To be tough" • To be unafraid of publicity
Self-organized working processes	Technical: • Dealing with technical problems Social: • Active communication skills • "E-mail-competency" Personal: • Intrinsic motivation to work in OS • Ability to learn • Openness to new things and approaches • Persistence • Time-Management • Ability to adapt to changing situations • Self-organisation • To demand high quality of own work • Curiosity
International Community	Social: • Intercultural competencies • English language skills • Dealing with different styles of communication
Explicit promotion to committer by other committers	Technical: • Identification of possible successful projects • Gaining recognition and earning reputation • High number of qualitative patches • Documentation of work Social: • Presentation skills • Ability to establish and maintain contact with the community Personal: • Motivation for participation in the community life • Internalisation of the "social give and take" philosophy of the community

criticism from other members of the community via mail (Interviews 1, 4, 5 and 6).

- Requirement of architectural competencies
Programming in OS is more than the knowledge of particular software. In order to find missing features or easily integrate them into new projects respectively working for common solutions in teams and respecting a bilateral feedback culture within the community (members), developers need an understanding of how systems are built (Interview 4). Otherwise they are not able to “think outside the box” (Interview 4).
- Building an honourable reputation
The rights of committers are connected with trust from the community and its clients. No community is searching for unpopular outsiders (Interview 2). Developers need to get visibility of OS members by asking relevant questions to maintainer (Interview 1), giving presentations on community conferences (Interview 1 and 2) or attending working groups in the community (Interview 1). The status can be approved by the public examination of the applicant’s work: “code talks” (Interview 1). That does not only mean to be competent and to do the right thing but to live the philosophy of social give and take as well as to show a social competent behaviour to other members (Interview 1).

The results illustrate the complexity of the successful developers profile and the need for a strategic planning of a career in OS. Applicants must take the chance to get into the right project at the right time and they need the competencies to be recognized and accepted to do so.

Asking the interviewees for demanded skills in the future the described competencies are mainly confirmed in their relevance. Technical competencies will be a stable aspect of a successful developer. But besides contributions documentary work is of increasing relevance as it is reflected in higher reputation for that kind of passive work. Social competencies are connected to a growing importance for future success in OS projects as users and developers are becoming more diverse in their cultural/linguistic background or personal expectations onto OS software. OS developers are not part of an inner circle of equals but members of a global community which includes users without technical experience. That means: Committers need to adapt products to new target groups. This makes English language skills, empathy, communication skills and the ability to provide constructive feedback necessary. The overall visibility makes a person necessary that is willing to fulfil the role of someone in the “publicity” of the community. Corresponding personal competencies are the ability to take feedback from others and compliance to the social rules of the community. Because of some cases of discrimination (e.g. against women) in the history of OS the community seems to be more sensitive to this part of work. Therefore aspiring developers should have these competencies in mind planning a career in OS.

5 Implications for Human Resource Management in OS Companies

5.1 The Need for a Comprehensive Competence Development Model in OS Software Companies

Building up a profile of the described competencies should not be only developer’s individual matter of question and time but also part of Human Resource Management in software companies: On the one side developers involved in important OS projects as a committer can become an economic resource trying to reach or maintain the company’s position in the market. The team-leaders and Human Resource managers of both companies interviewed emphasized the need for an approach to support their developers’ career inside the community as “clients connect the committer status with trust in the company’s competencies” (Interview 1). On the other side OS work unveils potential or force competencies of a developer that can be relevant for the company when the competencies are transferred to the workplace. Examples of these competencies that are connected with OS contribution are: fast induction into new software, involvement in current discussions and changes within the IT-market, sense of responsibility and being a communicative part of a team (Interviews 1–6).

As a conclusion software companies should be interested multiply in promoting their developers becoming a committer to get influence in relevant OS projects.

In order to realize competency developments which satisfy both the interests of the company and the developer Human Resource Management of software developers must be planned and implemented in a comprehensive way including the needs and motivations of the OS community, the developer and the company at the same time. The following chapter tries to summarize relevant principles to achieve this goal.

5.2 Elements of a Comprehensive Competence Development Model in OS Software Companies

Arranged around a common Human Resource Management model you can formulate the following implications for Human Resource Managers in OS software companies as an outlook. It systematizes the relevant aspects of planning and implementing Human Resource strategies in reference to the above mentioned competencies as a process. Successful strategies of Human Resource Management in the companies interviewed are integrated in order to illustrate the abstract recommendations for further research and practice.

• Demand Analysis

Within the demand analysis qualification requirements and potentials of the developers are identified in comparison with the OS goals of the company: Which competencies should be developed to maximise individual and economic success with the committer status? The process for this first step of Human Resource Management can be initiated by the developer

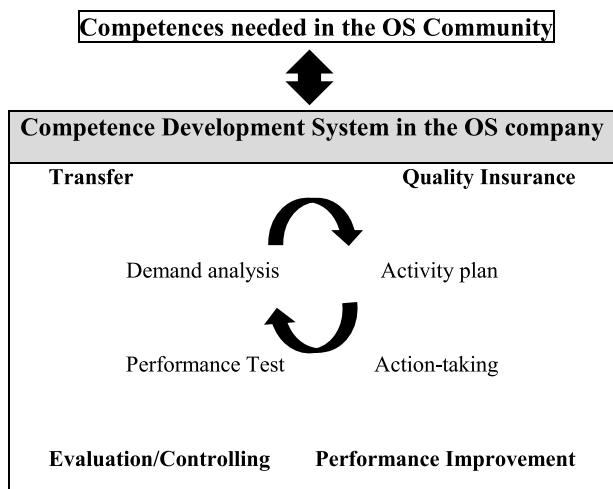


Figure 2 Elements of a competence development model in an OS company in order to foster developers' committer status in OS projects. **Origin:** Own figure, based on the Human Resource Management model of Stender [33].

or the company. Crucial for success is the involvement of the developer at a very early stage. Focus of the analysis can be the deficits or the potential of the developer.

- **Activity plan**

The activity plan must consolidate the motivation of the prospective participating developer. Competence development must be initiated intrinsic (Interview 5). For that the personal motivation of the developer needs to be considered. Considerations of the developers might be: "Is the time for participation in a Human Resource training equivalent to the effect towards my career?" (Interview 6). As the interviewees rated the intrinsic motivation of the OS software extremely high inducements to take part in a human resource activity should be connected with the personal goals the developer would like to achieve in the OS community (Interviews 1, 3, 4, 5 and 6). In order to choose the right projects connections to the community are compulsory (see Fig. 2). Existing committers in projects could be gateopener in informing the company about upcoming changes in projects (Interview 1). Otherwise developers need to get working time to analyse the market in order to find the most appropriate project (Interviews 3 and 6).

- **Action-taking**

Human Resource managers must be sensitized for possible and suitable ways of learning outside and within the OS community. Flexible arrangements of competence development including informal or social ways of learning (e.g. pair programming with colleagues) are suitable with regard to the mentioned relevant social competencies (Interview 2).

- **Performance Test**

With regard to the public documentation of OS activities this part of the model can be implemented

quite easily, e.g. by proofing the new behaviour of the developer on mailing lists in open source projects (Interviews 1, 4 and 6).

The sustainable success of these strategies is influenced by super-ordinated procedures shown in the outer circle of the four edges.

- **Transfer**

The competence development model is based on the idea that competences acquired as a committer in OS projects can be transferred to the workplace in the company. Further research will need to show how successful this transfer can be for various kinds of competencies. From Human Resource perspective transfer into the daily work life is influenced by the possibility to practice the new competencies in an adequate way and time. Managers can contribute to this sustainable development by a corresponding organisation of the team structure in consultation with the project managers. Developers should be encouraged to continue the OS activity in a specially reserved part of the working time. Continuous mentoring programmes with other committers in the company can be a supplementary strategy to patronize developers on their way to higher status in OS projects (Interviews 5 and 6).

- **Quality Management**

Human Resource strategies for developers are not to be autonomous from the strategic planning of the company. Therefore, criteria for success and improvement of the developer need to be specified before starting a competence development activity. Both sides – developer and company – should clarify their expectations of the committer status.

- **Performance Improvement**

Performance improvement takes into consideration any necessary structural or material support the developer needs to realize the career as committer in a particular OS project. But it also analyses possible aspects that hinder the developer to work with full potential before starting a corresponding activity. That is: Reasons for deficits in the competence profile of a developer might be caused by the working atmosphere in the company or the community itself and need to be analysed.

- **Evaluation/Educational Controlling**

This procedure underlies the whole process and embraces two parts: Satisfaction of the developer (evaluation) and economic effects for the company (educational controlling). Further research need to show how both aspects can be realized simultaneously and in which way the committer status of developers effect the company's return on investment. OS is an extraordinary working field when it comes to the question of measuring the competencies. As work itself is accessible to everyone in the community the contributions of everyone are also archived for open review by its members. "The net never forgets" (Interview 4). This public profile is not limited to technical

competencies. Social competencies can be observed by following the GitHub-Repositories and mailing-lists as well.

Acknowledgements

I would like to record my gratitude to Prof. Dr. Dirk Riehle from the University Erlangen-Nürnberg for his important part in this research project which was based on his interest in this topic. He was responsible for the sampling and contact to cooperating partners. Interviews were led together and the first coding was done in a pair-coding partnership. Besides he was a useful feedback partner. Also the comments of the reviewers were an excellent help to revise the article.

References

- [1] J. Bessen. Open Source Software: Free Provision of Complex Public Goods. In: *The Economics of Open Source Software Development*, ed. J. Bitzer and P. J. H. Schröder: 57–82. Amsterdam: Elsevier, 2006.
- [2] J. Bitzer, I. Geishecker, and P. J. H. Schröder. Returns to Open Source Engagement: An Empirical Test of the Signaling Hypothesis. Universität Oldenburg, 2010.
- [3] J. Bitzer, W. Schrettl, and P. J. H. Schröder. Intrinsic Motivation in Open Source Software Development. *Journal of Comparative Economics* 35(1): 160–169, 2007.
- [4] C. J. Choi, S. W. Kim, and S. Yu. Global Ethics of Collective Internet Governance: Intrinsic Motivation and Open Source Software. *Journal of Business Ethics*, 90: 523–531, 2009.
- [5] J. M. Corbin and A. C. Strauss. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. SAGE Publications, 3rd Version, 1998.
- [6] J.-M. Dalle and N. Jullien. ‘Libre’ software: turning fads into institutions? *Research policy*, 32(1): 1–11, 2003.
- [7] J. Erpenbeck and L. V. Rosenstiel. *Handbuch Kompetenzmessung*. Schäfer-Poeschel, 2003.
- [8] B. Frey and A. Stutzer. Should National Happiness be Maximized? Working Paper, University of Zurich, 2007.
- [9] I.-H. Hann. Why Do Developers Contribute to Open Source Projects? 2nd OSS Workshop, 2002.
- [10] A. Hars and S. Ou. Working for Free? Motivations for Participating in Open-Source Projects. *International Journal of Electronic Commerce*, 6(3): 25–39, 2002.
- [11] G. Hertel, S. Niedner and S. Herrmann. Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel. *Research Policy*, 32(7): 1159–1177, 2003.
- [12] P. Himanen. *The Hacker Ethic and the spirit of the Information Age*. New York: Random House, 2001.
- [13] J. P. Johnson. Open Source Software: Private Provision of a Public Good. *Journal of Economics and Management Strategy*, 11(4): 637–662.
- [14] B. Kogut and A. Metiu. Open-source software development and distributed innovation. *Oxford Review of Economic Policy*, 17: 248–264, 2001.
- [15] K. Lakhani and E. Von Hippel. How Open Source Software Works: “Free” User-to-User Assistance. *Research Policy*, 32(7): 922–943, 2003.
- [16] K. Lakhani and R. G. Wolf. *Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects*. Boston: MIT Sloan Working Paper no. 4425–03, 2003.
- [17] G. Lee and R. E. Cole. From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development. *Organization Science*: 14: 633–649, 2003.
- [18] J. Lerner and J. Tirole. The simple economics of Open Source. Working Paper 7600, National Bureau of Economic Research, 2000, online: <http://www.nber.org/papers/w7600>.
- [19] J. Lerner and J. Tirole. Some Simple Economics of Open Source. *Journal of Industrial Economics* 50, 2: 197–234, 2002.
- [20] T. Mikkonen, T. Vaden, and N. Vainio. The Protestant ethic strikes back: Open Source Developers and the ethic of capitalism. *First Monday*, Volume 12, 2, February 2007.
- [21] A. Mockus, R. T. Fielding, and J. D. Herbsleb. Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology*, 11: 309–346, 2002.
- [22] J. Y. Moon and L. Sproull. Essence of distributed work: The case of the Linux kernel. In P. Hinds & S. Kiesler (Eds.), *Distributed work: New research on working across distance using technology*: 381–404. Cambridge, MA: MIT Press, 2002.
- [23] J. Matusow. Shared Source: The Microsoft Perspective. In J. Feller, B. Fitzgerald, S. A. Hissam, and K. R. Lakhani (Eds.): 329–346. Cambridge, MA, USA: The MIT Press, 2005.
- [24] D. P. Myatt and C. Wallace. Equilibrium Selection and Public Good Provision: The Development of Open-Source-Software. In. *Oxford Review of Economic Policy* 18, 4: 446–461, 2002.
- [25] S. C. O’Mahony and F. Ferarro. The Emergence of Governance in an open source community. *Academy of Management Journal*, 50(5): 1079–1106, 2007.
- [26] S. C. O’Mahony, Nonprofit Foundations and Their Role in Community-Firm Software Collaboration. In J. Feller, B. Fitzgerald, S. A. Hissam, and K. R. Lakhani (Eds.), (p. 393–414). Cambridge, MA, USA: The MIT Press, 2005.
- [27] S. C. O’Mahony. Guarding the Commons: How Community Managed Software Projects Protect Their Work. *Research Policy*, 32(7): 1179–1198, 2003.
- [28] E. S. Raymond. Homesteading the Noosphere: An Introductory Contradiction. 1999. Online: <http://tuxedo.org/~esr/writings/homesteading> (accessed March 1 2013).
- [29] D. Riehle. A New Developer Career. Linux-Tag 2010. 2010. Online: <http://dirkriehe.com>.
- [30] S. Sharma, V. Sugumaran, and B. Rajagopalan. A framework for creating hybrid-open source software communities. *Information systems Journal*, 12(1): 7–25, 2002.
- [31] R. Stallman. The GNU Operating System and the Free Software Movement. In. Chris DiBona, Sam Ockman, and Mark Stone, editors, *Open Sources: Voices from the Open Source Revolution*. Sebastopol, California: O’Reilly: 53–70, 1999.
- [32] D. Stewart. Social status in an open source software community. *American Sociological Review*, 70: 823–842, 2005.
- [33] J. Stender. *Betriebliches Weiterbildungsmanagement*. Hirzel, Stuttgart, 2009.
- [34] J. D. Thompson. *Organizations in action*. New York: McGraw-Hill, 1967.
- [35] L. Torvalds and D. Diamond. *Gewoon voor de fun (Just for Fun)* (C. Jongeneel, Trans.). Uithoorn: Karakter Uitgevers, 2001.
- [36] R. van Wendel de Joode, L. Yuwei, and D. Shay. Rethinking Free, Libre and Open Source Software. *Knowledge, Technology, & Policy*, 18(4): 5–16, Winter 2006.
- [37] E. Von Hippel and G. Von Krogh. Open Source Software and “Private-Collective” Innovation Model: Issues for Organization Science. *Organization Science*, 14(2): 209–223, 2003.
- [38] G. V. Krogh, S. Spaeth, and K. R. Lakhani. Community, joining, and specialization in open source software innovation: A case study. *Research Policy*, 32: 1217–1241, 2003.
- [39] S. Weber. *The Success of Open Source*. Cambridge: Harvard University Press, 2004.

Received: March 6, 2013

**Prof. Dr. Nicole Kimmelman**

since 04/11 Juniorprofessor for Professional Competence Development (University Erlangen-Nuremberg)

08/06–04/11 Research Assistant at the Chair for Vocational Education and Training at University Erlangen-Nuremberg (2009: Phd degree Dr. rer.pol)

03/06–08/06 Teacher for German as a foreign Language

10/05–03/06 Assistant of the National German Advisor, New Zealand, Wellington

10/04–09/05 Teacher /Educational Manager

09/05 Degree in German as a Foreign Language (Certificate)

09/04 Degree in VET & Human Resource Management (Diploma)

09/97–07/99 Apprenticeship as a bank clerk

07/97 High School degree

1978 Born in Nuremberg

Address: Juniorprofessur für Berufliche Kompetenzentwicklung, Universität Erlangen-Nürnberg, Lange Gasse 20, D-90403 Nürnberg, e-mail: nicole.kimmelman@wiso.uni-erlangen.de

Preview on issue 6/2013

The topic of the next two issues prospectively will be “Engineering Adaptive Software Systems” (Guest Editors: N. Ritter and W. Renz) and “Security in Business Processes” (Guest Editor: R. Accorsi). They will contain following papers:

“Engineering Adaptive Software Systems”

- *K. Herrmann et al.*: Online Horizontal Partitioning of Heterogeneous Data
- *W. Powley and K.-U. Sattler*: A Framework for Autonomic Workload Management in DBMSs
- *St. Baumann and K.-U. Sattler*: Autonomic Physical Database Design – From Indexing to Multidimensional Clustering
- *K. Geihs et al.*: On the Development of Self-Adaptive Socially Aware Ubiquitous Computing Applications

“Security in Business Processes”

- *A. D. Brucker*: Integrating Security Aspects into Business Process Models
- *G. Governatori*: An Abstract Normative Framework for Business Process Compliance
- *N. Zannone et al.*: Privacy Analysis of User Behavior Using Alignments
- *F. Böhr et al.*: Business Process Security Analysis – Design Time, Runtime, Audit Time



Oldenbourg
Verlag

Ein Unternehmen von De Gruyter

Das Standardwerk zur IT-Sicherheit – umfassend, praxisrelevant, aktuell



Claudia Eckert
IT-Sicherheit
Konzepte – Verfahren – Protokolle

8. Auflage 2013
XIV, 1016 Seiten
Flexcover
ISBN 978-3-486-72138-6
€ 69,80

Informationstechnologie (IT) ist heute in nahezu allen Bereichen von zentraler Bedeutung. Ergebnisse von Sicherheitsanalysen und die wachsende Zahl an Angriffen auf vernetzte IT-Systeme verdeutlichen die großen Sicherheitsprobleme, die mit diesen komplexen IT-Systemen einhergehen.

In diesem Buch werden die zur Umsetzung der Sicherheitsanforderungen benötigten Verfahren und Protokolle detailliert vorgestellt und anhand von Fallbeispielen erläutert, so dass nicht nur ein Bewusstsein für IT-Sicherheitsrisiken entwickelt werden kann, sondern auch konkrete Gegenmaßnahmen ergriffen werden können.

Ein Muss für jeden, der sich mit dieser hochaktuellen Problematik beschäftigt!



Prof. Dr. Claudia Eckert ist Inhaberin des Lehrstuhls Sicherheit in der Informatik der TU München und Direktorin der Fraunhofer-Einrichtung für Angewandte und Integrierte Sicherheit (AISEC) mit Sitz in Garching bei München.

Bestellen Sie in Ihrer Fachbuchhandlung
oder direkt bei uns:

www.degruyter.com/oldenbourg