



# The Unstoppable Rise of Open Source

## Der Siegeszug von Open Source

Dirk Riehle\*, Friedrich-Alexander-Universität Erlangen-Nürnberg

\* Correspondence author: [dirk@riehle.org](mailto:dirk@riehle.org)

Open source software is software that is available for everyone to use for free, to adapt to one's needs for free, and to pass on to other interested parties, for free as well [1]. Beyond the legal definition, the open source movement has brought the world a new way of developing software. Open source is a "development method for software that harnesses the power of distributed peer review and transparency of process. The promise of open source is better quality, higher reliability, more flexibility, lower cost, and an end to predatory vendor lock-in." [2]

Underlying open source software development are the principles of open transparent collaboration: egalitarianism (everyone may participate, nobody is excluded a priori), meritocracy (decisions are based on the merits of the arguments) and self-organization (processes are adapted to people and project communities rather than people and project communities to processes) [3].

Recent research has shown that much of open source software development is commercially sponsored software development: Open source has arrived in the mainstream [4].

This special issue on open source shows how open source has become an economically sustainable phenomenon (articles 1 and 2), how it works (articles 3 and 4), and how to engage in it to have a career (article 5).

The first article by Jesus M. Gonzalez-Barahona and Gregorio Robles, both of Universidad Rey Juan Carlos, illustrates the rich history of open source software, describes a model of how companies engage with open source communities, and provides several successful examples of such engagements. A model of community-company relationship governance is presented and discussed. This model and the examples

should help anyone trying to understand how their company could engage with an existing open source community or how they should think about setting up a new open source project and its community, if this was their goal.

The second article by Anthony I. Wasserman of CMU Silicon Valley looks at business models of companies developing or sponsoring open source software projects. How can they generate significant revenue if they are giving their intellectual property away for free? Wasserman finds answers in different business models, built both on community open source and commercial open source. Mostly focused on revenue streams, Wasserman observes that companies earn a living from software subscription fees, application hosting, commercial licenses, documentation and training, complementary products like hardware, and good old merchandising. In a way, what is not surprising is what companies receive revenue for: It is very similar to the revenues that traditional closed source companies generate. The difference to traditional companies is in the business strategies, including software production, as discussed by other articles in this special issue.

The next article by Stefan Koch of Bogazici University focuses on open source communities and the work results they produce. In reviewing prior work Koch finds that software quality increases with growing project size but that at the same time the projects should be split into smaller component teams. In other related work, the author finds that open source software projects are significantly more efficient than traditional software development projects. The author proceeds to present original work on how open source communities adopt



new processes and practices. Not surprisingly, he finds that the adoption of new tools and more formal practices is mostly driven by the size of the development team, validating the hypothesis that even open source projects, as they grow larger, need and choose a more structured process, than may be necessary for smaller projects.

The fourth article by Wolfgang Mauerer and Michael C. Jäger of Siemens Corporate Technology then discusses the open source software engineering process. Open source engineering is publicly visible work across company boundaries, it takes place in small incremental steps and creates new roles in programming. A multitude of requirements follows from the open collaborative nature of open source engineering processes that make it different from traditional in-house software development. For example, a staged programming and code review process as well as the pressure to not to embarrass oneself publicly leads to the high quality of open source code observable in popular projects. Also, the authors argue, the highly collaborative and distributed nature of open source software development has given us the new breed of distributed configuration management systems that are now poised to replace the prior generation of centralized configuration management systems.

The fifth and final article by Nicole Kimmelman of Friedrich-Alexander-Universität Erlangen-Nürnberg reviews the competencies a software developer should have for a successful career in open source. Looking at technical, social, and personal competencies, Kimmelman finds that developers certainly need technical abilities like good programming skills, but beyond that need to have significant social skills to advance in an open source project. Given the international and distributed nature of open source software development, good English skills are a required personal competence as well. The article provides many more competencies and concludes with a recommended career development model that human resource departments of software companies can pick up.

Open source has given the world more than business models and development methods for software

companies. Beyond software, the principles of open collaboration are being found in or applied to wikis and Wikipedia, open data and open government, open access, citizen engineering, open educational resources, open innovation, and many more. OpenSym, a new ACM-supported conference on “everything open”, is capturing this spirit and is providing a forum and community for researchers and practitioners of open collaboration [5]. OpenSym 2014 will take place in Berlin, Germany in August 2014.

We haven’t see the end of it yet; rather, we are just at the beginning.

## References

- [1] See <http://opensource.org/osd>.
- [2] See <http://opensource.org/about>.
- [3] See <http://www.opensym.org/about/definition/>.
- [4] See <http://lwn.net/Articles/222773/>.
- [5] See <http://opensym.org/>.



**Prof. Dr. Dirk Riehle**, M. B. A., is the Professor of Open Source Software at the Friedrich-Alexander-Universität Erlangen-Nürnberg. Before joining academia, Riehle led the Open Source Research Group at SAP Labs, LLC, in Palo Alto, California (Silicon Valley). Riehle founded the Wiki Symposium, and more recently, the Open Symposium, now the joint international conference on open collaboration. He was also the lead architect of the first UML virtual machine. He is interested in open source software engineering and agile methods, complexity science and human collaboration, and software design. Prof. Riehle holds a Ph.D. in computer science from ETH Zürich and an M.B.A. from Stanford Graduate School of Business. He welcomes email at [dirk@riehle.org](mailto:dirk@riehle.org), blogs at <http://dirkriehle.com>, and tweets as @dirkriehle.

Address: Friedrich-Alexander-Universität Erlangen-Nürnberg, e-mail: [dirk@riehle.org](mailto:dirk@riehle.org)