

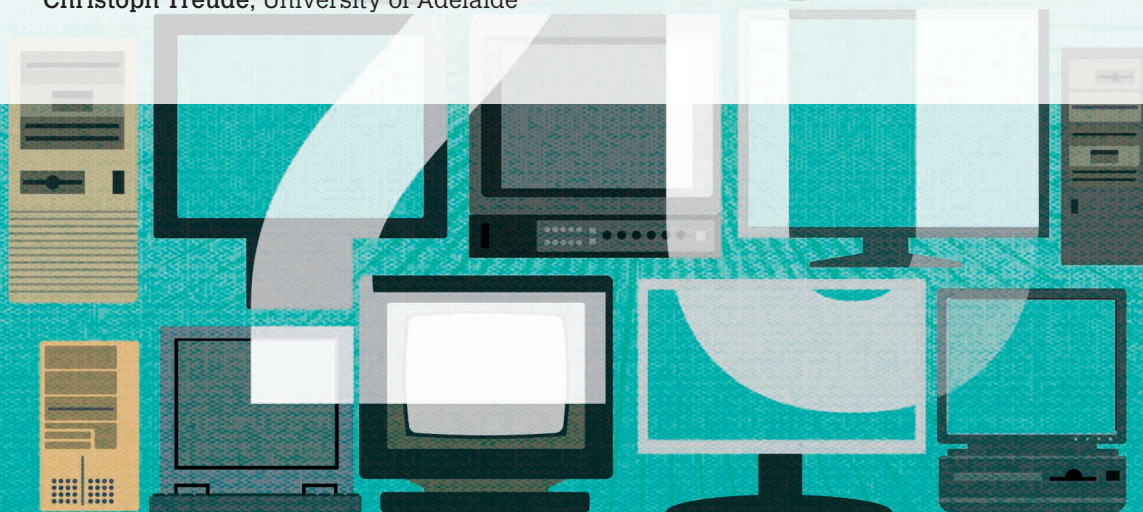
# Twenty Years of Open Source Software: From Skepticism to Mainstream

Gregorio Robles, Universidad Rey Juan Carlos

Igor Steinmacher, Northern Arizona University

Paul Adams, Wayfair

Christoph Treude, University of Adelaide



# YEARS

OF OPEN SOURCE

**OPEN SOURCE SOFTWARE** (OSS) has conquered the software world. You can see it nearly everywhere, from Internet infrastructure to mobile phones to the desktop. In addition to that, although many OSS practices were viewed with skepticism 20 years ago, several have become mainstream in software engineering today: from development tools such as Git to practices such as modern code reviews.

In the programmer community, OSS has become so prevalent that some companies now expect potential employees to have an active GitHub profile that showcases their OSS contributions. For a phenomenon with as much impact on software development practice as OSS, it is essential that we understand what works, what does not, and why. Although researchers and practitioners have studied OSS for many years, we still do not have a complete understanding of OSS as a whole or the many aspects related to it.

Akin to the famous quote about Wikipedia: “The problem with Wikipedia (read OSS) is that it only works in practice. In theory, it can never work,” we see OSS impacting our lives every day, yet there is only a very limited number of theories about OSS that describe, explain, or predict how OSS impacts software engineering practice.

The goal of this *IEEE Software* theme issue is to share with software engineering practitioners the reports that analyze those OSS processes, practices, and tools that have had a major influence on software engineering practice.

Topics of recent interest for academia and practitioners include legal aspects (the roots of OSS), development tools and platforms, programming languages, global software development, software ecosystems,

social software development, socio-technical aspects, software analytics, software heritage, innersource, and modern code review, among others.

Nonetheless, OSS has spread its philosophy well beyond software engineering and inspired many other movements and initiatives, such as open innovation, open hardware, open government, open content (e.g., Wikipedia and OpenStreetMap), and open educational resources. Even the way researchers publish their research has changed, with many attempting to have their publications available under open access and following open science principles. For instance, *IEEE Software* is currently cataloged as a green open access journal by SHERPA/RoMEO, a service that provides the copyright and open access self-archiving policies of academic journals. Green open access allows an author to archive a preprint or postprint version of his or her paper and make it publicly available. The archive can be located on the personal home page of the author, at an institutional repository of his or her employer, or at an e-print server such as arXiv.

Given the importance that OSS has gained in and beyond software engineering practices, we argue that it is important for the software engineering community to benefit from the insights of having an overview of the realities, promises, generalizations, and pitfalls of OSS.

### Is Open Source Really 20 Years Old?

The answer is: not exactly. What has just become 20 years old is the “Open Source Definition,” the first version of which was published in 1998 by notable members of the open source community. The “Open Source Definition” is a document

that lists 10 conditions software licenses must comply with to be considered open source. This definition is an adapted version of the Debian Free Software Guidelines (DFSG) primarily authored by Bruce Perens, with feedback from other Debian developers and released in mid-1997.

The DFSG is a set of guidelines used by the Debian community to determine what constitutes “free software” and to help make decisions about what software they include in their distribution. These guidelines were a more detailed implementation of the concept of free software that Richard M. Stallman envisioned with his four freedoms (initially only three, but then freedom zero was added), although the authors of DFSG did not know about the latter (i.e., the four freedoms) because these had been published in an early edition of the *GNU’s Bulletin*, which was mostly distributed in paper form on the Massachusetts Institute of Technology campus in 1986. It was not, however, actively promoted on the Free Software Foundation’s website as an alternative to the “Open Source Definition” in the late 1990s.

Stallman claims that, in its early days, software was de facto open source (he refers to it as *free*) but that the software industry in the late 1970s and 1980s based its business model on applying copyrights in a very restrictive way. Although we acknowledge the influence of early programmers and the ethical side of the free software movement, the scope of this issue as well as a major part of software engineering research, is the effort of those who promoted the term *open source* to be more business friendly and practical.

OSS is an evolving environment that has changed considerably in the last 20 years. Scholars refer to three





**GREGORIO ROBLES** is an associate professor at the Universidad Rey Juan Carlos, Madrid, Spain. His research focuses on free/open source software. Robles received his Ph.D. in empirical software engineering research on free/open source software in 2006. He coauthored the 2002 FLOSS study while working at the University of Maastricht, The Netherlands. He was the general chair of the IEEE International Conference on Software Maintenance and Evolution in 2018. He was also the program cochair of the 2016 and 2017 Open Source Software Conferences, among others. He is a Member of the IEEE. Contact him at [grex@gsyc.urjc.es](mailto:grex@gsyc.urjc.es).



**PAUL ADAMS** is a senior engineering manager at Wayfair as well as an advisor to Swarm64 on open source. His focus is on open source community productivity. Adams received his Ph.D. in open source community productivity from the University of Lincoln, United Kingdom. Contact him at [padams3@wayfair.com](mailto:padams3@wayfair.com).



**IGOR STEINMACHER** is an assistant professor in the School of Informatics, Computing, and Cyber Systems at Northern Arizona University, Flagstaff. His research interests include human aspects of software engineering, behavior in open source software communities, mining software repositories, and software engineering education training. Steinmacher received his Ph.D. in computer science from the University of São Paulo, Brazil. He served as program cochair of the 12th International Workshop on Cooperative and Human Aspects of Software Engineering, was a guest editor of a special issue on open source software for *Journal of Internet and Software Applications*, and he is currently the program cochair of the 15th Association for Computing Machinery/IEEE International Conference on Global Software Engineering 2020. He is a Member of the IEEE. Contact him at [.steinmacher@nau.edu](mailto:.steinmacher@nau.edu).



**CHRISTOPH TREUDE** is a senior research fellow in the School of Computer Science at the University of Adelaide, Australia. The goal of his research is to advance collaborative software engineering through empirical studies and the innovation of processes and tools that explicitly take the wide variety of artifacts available in a software repository into account. Treude received his Ph.D. in computer science from the University of Victoria, Canada. He currently serves on the editorial board of *Empirical Software Engineering* and as general cochair of the IEEE International Conference on Software Maintenance and Evolution 2020. He is a Member of the IEEE. Contact him at [christoph.treude@adelaide.edu.au](mailto:christoph.treude@adelaide.edu.au).



generations of OSS: the first one in which a community of volunteers led the development process, a second one in which the software industry began to interact with the community, and a third one in which industry consortia push OSS projects forward with the help of a significant amount of professional (i.e., paid) developers. Because these three generations coexist, community-driven and highly industrialized projects are led by a consortium of companies that are not necessarily the traditional IT sector. In many sectors, the leading product is OSS. And if it is not, its major competing product is.

### In This Issue

In “How to Succeed in the Software Business While Giving Away the Source Code,” Brosgol explores the challenges of allowing users to access, modify, and redistribute the source code while protecting a company’s intellectual property and sustaining a growing and profitable revenue stream—using AdaCore’s experience with the open source licensing for its major product line, the GNAT Ada development toolset. The author describes the rationale for AdaCore’s open source approach, the relationship between the company and the open source developer community, and identifies issues and how they have been addressed, along with valuable lessons learned.

In “From Art to Science: The Evolution of Community Development,” Mueller and Izquierdo-Cortazar emphasize the importance of community development efforts in maintaining and nurturing the health and well-being of open source ecosystems. The article showcases the results of applying a data-driven network analysis approach to the OpenShift and Cloud Native Computing Foundation

communities, with a particular focus on the Kubernetes community. The authors explore the structure and stability of networks, identify and promote collaborations, and develop strategies to improve ecosystem health and stability, with the ultimate goal being to provide a holistic understanding of the ecosystems of projects and organizations.

In “Critical Factors for Open Source Advancement in the U.S. Department of Defense,” Scanlon describes the critical success factors needed for successfully leveraging OSS components in U.S. Department of Defense software systems. The five factors that are identified in the article (i.e., knowing the supporting policies, making informed selections of open source components, addressing security concerns, leveraging trusted resources, and understanding licensing issues) are applicable to other fields and industries where open source use is still met with skepticism and resistance and can support open source initiatives in any environment.

### Looking Forward

In the current environment, there is increasing interest in OSS from a practitioner’s perspective, noticeable not only by the number of paid developers but also by the rising number of previously closed source projects that have been open sourced.

Recently, the rising importance of software in modern society has inspired many industrial sectors to embrace OSS. Interestingly, companies that have historically been fierce competitors are now cooperating to develop software products for the marketplace. For these companies, OSS has been instrumental in making this cooperation possible. Beyond the production of software, key concepts of OSS, such as community and sharing, are now a relevant aspect of their innovation strategy. When

senior management wants to implement OSS, however, there is still a lack of formal theories and evidence. OSS relies too much on stories bordering on mythical, such as the one of a computer science student creating an operating system from scratch with the help of the Internet and competing against the biggest software company in the world.

The IEEE Bureau of Labor Statistics defines *software engineering* as “the systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software.” As in many other areas of software engineering, with OSS, we have some partial scientific and technological knowledge as well as methods and experience but must further investigate them to apply OSS systematically. It is in this regard that we hope this special issue has contributed to this endeavor while helping practitioners and academics reflect on all of these aspects. We look forward to seeing continued research in this area.

**W**e sincerely thank the writers and reviewers of all of the high-quality submissions we received for this theme issue. We also thank Editor in Chief Ipek Ozkaya and her *IEEE Software* crew for its guidance and support. Happy reading. ☺

### Acknowledgments

Gregorio Robles is supported by the Madrid Regional Government (e-Madrid-CM—P2018/TCS-4307), cofinanced by E.U. Structural Funds (FSE and FEDER), and by the Spanish Government (RTI2018-101963-B-I00). Christoph Treude is supported by the Australian Research Council’s Discovery Early Career Researcher Award funding scheme (DE180100153).