

How Open Source Is Changing the Software Developer's Career

Dirk Riehle, Friedrich-Alexander-Universität Erlangen-Nürnberg

Software developers with open source project experience acquire verifiable technical expertise, peer-certified competencies, and positional power—advantages that align with companies' need to obtain a competitive advantage.

Open source software development is adding skills to the labor market and offering the promise of increased salaries and job security to those with open source project experience. This trend is not surprising, given that open source software development has long overlapped with the commercial world. For example, a 2012 Linux Foundation study found that more than 75 percent of the Linux kernel is being developed on company time or with company sponsorship (<http://go.linuxfoundation.org/who-writes-linux-2012>). A more recent independent study¹ found that about half of all code contributions are being performed Monday to Friday between 9 am and 5 pm, likewise suggesting paid work. Clearly, this growing commercialization of open source is influencing software developer careers.

Moreover, the vast majority of successful open source projects are community owned,² and community open source software is embedded in nearly all commercially relevant software today, open or closed source.

The Apache webserver and Firefox browser are notable examples. Thus, most software products are building on community open source software; Apache Cloud-Stack is a case in point.

Because of this embedding, the importance of open source for innovation and the software industry cannot be underestimated:

a startup company can now launch a prototype relatively cheaply using free software and services. An example is the rapid growth of Sonatype's Central Repository, which served nearly 8 billion requests for Java open source components in 2012 (www.sonatype.com/news/sonatype-secures-access-to-the-central-repository-for-component-based-software-development). With this growing dependence on open source software, companies increasingly need and want to influence the direction of its development. Software developers in important positions of economically relevant open source projects are highly sought after in the labor market.

To determine how open source is affecting software development as a career, I interviewed practitioners via email from 2011 to 2012³ and reviewed pertinent literature. My investigation revealed that respected open source software developers often enjoy higher salaries, greater job security, and a richer work experience. Their competitive edge stems from their verifiable skills,

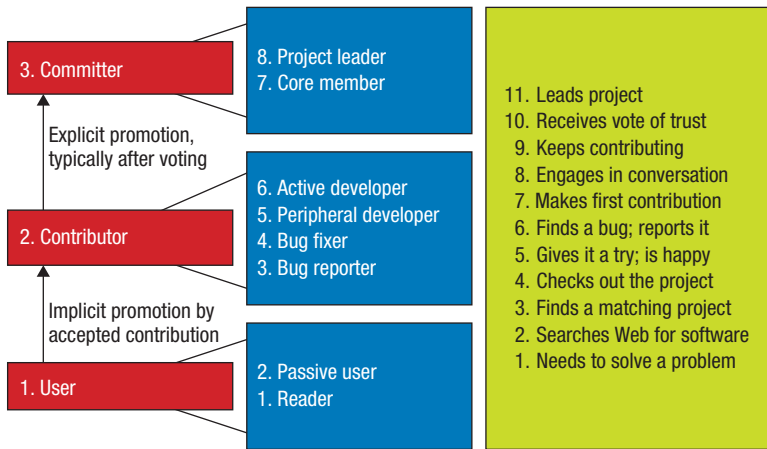


FIGURE 1. Three models of the open software development career ladder. The three roles on the left represent the simplest progression model. The center boxes show roles from the Onion Model. The box on the right builds on the Onion Model with individual tasks defined by Brian Behlendorf, Apache's primary developer.

peer-certified competencies, and positional power. At the same time, the open source movement is leveling the playing field and reducing labor market entry barriers, making life more difficult for those software developers who lack open source competence or status.

Erich Gamma, co-recipient of the 2011 ACM Software System award for his work on the open source Eclipse platform, related his experience in hiring software developers:

... when I received the first job application with a link to a code contribution to an open source project, I immediately followed the link, reviewed the code, invited the candidate for an interview round, and eventually made an offer. A link to a code contribution to an open source project is a great differentiator in a job application.

That said, participation in open source software development need not limit traditional pursuits. More often than not, it enhances the developer's corporate career and provides new work opportunities.

A NEW CAREER LADDER

In a traditional software development career, a developer enters the labor

market and eventually finds employment at some company. A mature enterprise defines the developer's career ladder, and a step upward typically implies increased seniority, power, and salary. The developer might evolve a technical career into engineering or product management, or choose something entirely different.

Behind this corporate ladder, career steps in open source software development have emerged based primarily on the status the developer enjoys in one or more open source projects. Although distinct, the career ladders can be mutually supportive; skills gained in a traditional career are applicable to open source projects, and the reputation and status gained in open source projects can help the developer advance within the company.

Role progression

Figure 1 shows an individual's career path in open source software development. The simple three-role model (left) starts with a user, any individual (including a developer) who uses the software. As its name implies, a contributor is someone who contributes to the project, for example, by submitting a bug report or a code patch. A committer decides which contributions should be included in the project.

The Onion Model⁴ (center) subdivides these roles into eight categories based on participants' proximity to the open source project. People start outside the project as readers and passive users; then move to the project periphery as bug fixers, bug reporters, peripheral developers, and active developers (developers without commit rights); and finally enter the project as core members and project leaders. Other models more explicitly define these career steps.⁵ For example, Brian Behlendorf⁶ lays out a complex role migration process built on the Onion Model that specifies 11 tasks (Figure 1, right).

In a career ladder analysis, the focus must be on one's formal status within the open source project and power to determine its content, scope, and direction. Individual competencies to become an open source software developer³ are less important than ably fulfilling the roles of user, contributor, and committer.

User and contributor. Because open source software is free, anyone can be a user. The number of contributors is more restricted: a bug report or patch submission must be accepted as relevant before it can become a contribution, and a typical project rejects many patch submissions or pull requests (code contributions) as irrelevant.⁷ Achieving contributor status is thus an implicit promotion, albeit an unheralded one since no one but the contributor notices the role change.

Both the user and contributor roles come with little power, since they only involve requesting the project to make a particular choice such as to add functionality or decide on a supporting technology.

Committer. The committer role carries considerably more power. Committers are named for the commit bit in the project's configuration management system (code repository). When the bit is set to true, that user is free to change the code base without permission. Users and contributors cannot write to the code repository; their commit bit has not been set. A committer can decide whether or not to include a contributor's patch or pull request—quality assurance responsibility and power rolled into one.

Moving from contributor to committer is an important career step. Most mature projects restrict committer status to those with proven loyalty to the project, typically measured by prolonged activity as a contributor. In most cases, existing committers discuss a proposed committer's merits and ultimately vote on changing that individual's status. The Apache Software Foundation (ASF) and Eclipse Foundation have even codified most of the committer election process and, once committers have arrived at a decision, the foundation publicizes it, sometimes with great fanfare.

The advent of decentralized configuration management systems and services, notably GitHub and Atlassian's Bitbucket, has sharpened the definition of committer from the arcane "someone with the commit bit set" to trusted project source. The traditional contributor-to-committer career step, however, remains an important event in a project, particularly those run by open source foundations.

In short, committers help lead a project and determine its direction. They perform important quality assurance and are indispensable in rallying contributors around project goals and

motivating them to pick up development work.

From roles to foundations

In response to the rapid growth of commercially relevant open source software, foundations have evolved to ensure the stability of such projects—that they are valid intellectual property and evolve collaboratively⁸—and to protect their developers from legal challenges. As such, these foundations have extended the open source career beyond the committer role, adding stages and expanding management status. As Figure 2 illustrates, developers with committer status can join project management committees (PMCs), become PMC leaders, and even achieve foundation membership.⁶

Before open source foundations, projects evolved without formal authority. Open source foundations now coordinate previously independent projects and start new ones. The ASF, Eclipse Foundation, and Mozilla Foundation coordinate many projects under their respective auspices to ensure that the projects are working together smoothly to form one or more viable software platforms.

Justin Erenkrantz, a former ASF president, remarked:

At the ASF, PMC members are recruited from project contributors. As recognized project stewards, all PMC members, including the appointed chair, wield significant power over the project through veto power.

This need for coordination has led to the creation of additional career steps, extending the committer role

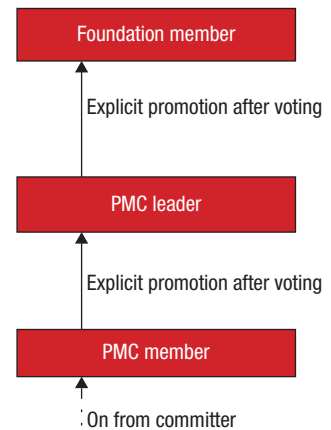


FIGURE 2. Extended open source software development career model. Open source foundations add to the user, contributor, and committer roles in the basic career model. A committer can move on to become a project management committee (PMC) member, a PMC leader, and eventually a foundation member.

to management. In the role's original scope, the committer reviews contributions and enters them in the code repository. As a PMC member, the developer helps determine the roadmap for one or more projects. As a PMC leader or one of several leaders, if a management committee has oversight for multiple projects, the developer gains even more power and influence.

The details of these final career steps vary by foundation, since only a few individuals can assume a PMC member or leader role. However, the steps always signify increasing power and influence. In the basic career model a committer determines a single project's direction, but in the extended model a developer in the final foundation member stage (management role) might influence an entire industry platform.


OPEN SOURCE COMPETENCIES AND STATUS

A software developer's open source activities serve as a signal to prospective employers that the developer has a certain collection of competencies and development status, including verifiable claims to technical skills and peer-certified competencies as

well as demonstrated influence in a particular project.

Verifiable technical skills

An open source software developer performs work for everyone to see, showcasing to the public technical skills that are independent of a particular project and documenting them for any interested party.⁹ This transparency is in sharp contrast to closed source work, which no one but the employer can view.



WEBSITES AND SERVICES MAKE IT
EASY TO A EVALUATE A DEVELOPER'S
REPUTATION AND OPEN SOURCE WORK.

The developer also has ready evidence of skills related to a particular project. A hiring manager can go to the project's website and see the developer's expertise firsthand—a freedom not possible when evaluating traditional software developers, whose work is behind the previous employer's firewall.

Remarks from Chris DiBona, director of Google's open source programs, support this hiring advantage:

Open source software is strategic to Google, and naturally we hire a great number of open source developers. Someone who demonstrates their ability by contributing to open source projects shows that they are able to code in the real world in ways other developers cannot readily match. It's the ultimate referral.

A host of websites and services make it easy to evaluate a developer's reputation and open source work. For example, Black Duck Software's OpenHub.net provides a comprehensive assessment of a software developer's open source activities and how other developers relate to that individual.

Peer-certified competencies

A sufficiently large, well-working open source project validates a developer's competencies simply by making the

developer one of its own. A hiring manager can gauge technical, social, and leadership skills through peer certification.

Technical. A project that includes the developer's work makes that work more valuable, essentially elevating the developer from user to contributor. The developer is no longer just performing work in public, but rather has demonstrated peer acceptance—an informal seal of approval from project colleagues.

Social. If a contributor receives a vote of trust and is promoted to committer status, the existing team of committers is certifying not only that the developer can work on a team but is desirable enough to work on *their* team. Thus, achieving committer status in a large, successful open source project validates a developer's social skills.

Leadership. A developer who becomes a committer, or better yet a PMC member, signals the makings of a project leader worthy of trust to inspire project members to persevere in meeting goals.

Robert O'Callahan, a distinguished engineer at Mozilla, commented on the value of open source contributions to corporate culture:

Open source contributors tend to believe in and practice the values that characterize successful open source projects, such as community, meritocracy, and transparent government. Hiring those people strengthens those values within your corporate culture.

Thus, hiring open source developers can also be a means for embracing or strengthening corporate values of collaborative software development. This applies not only to open source software but also to firm-internal or inner-source software, where it can help tear down development silos.¹⁰

Position of influence

In achieving committer status, a developer assumes significant formal power. Not only can committers insert code without peer review, but they also can influence the project's tone, content, and strategic direction, including architectural decisions.

Such influence can be used within the project to lead and inspire collaboration, and it can also be projected externally—for example, to keep competing developers out of the project by causing their bid for committer status to be rejected.

The committer becomes visible within the community and often

A COMPANY THAT HIRES A DEVELOPER WITH PROVEN SUCCESS IN OPEN SOURCE WORK ATTRACTS OTHER COMPETENT APPLICANTS.

beyond it. As a committer, a developer might become the project voice and be asked to publish project details or speak at conferences, for example.

NEGOTIATION POWER

Open source competency is a powerful negotiation tool for prospective employment. Companies look favorably on open source project participation, which can lead to higher salaries, greater job security, and a richer work experience.

Value of verifiable technical skills

As a general hiring rule, an applicant's skill set is commensurate with the salary the applicant can negotiate. However, one study has empirically validated that ASF committers have a higher salary than those with equivalent skills but no committer status.¹¹ Another study¹² found no such increase when looking at a broad array of open source projects, although most were not commercially relevant.

Even so, employers tend to give higher salaries to individuals who present less of a hiring risk. Managers who can view an applicant's sustained open source work will be more certain of the degree and nature of that individual's technical skills. Such confidence reduces the uncertainty discount that is part of any salary negotiation.

Comments from Marten Mickos, CEO of Eucalyptus and former CEO of MySQL, support this point:

From a software vendor's perspective, open source work on a developer's resume ... shows that the developer has a genuine passion for writing software and a level of self-confidence. ... If

developers even contributed to our [open source] products, ... ramp-up time will be shorter and we know they are likely to be a better fit than an unknown developer. All of this leads us to prefer open source developers when hiring.

If the technical skills are from a project of commercial interest to the employer, the developer can expect even greater salary negotiation power. The value of verifiable technical skills accrues to contributors, committers, and PMC members and leaders alike, and is by far the most common benefit of open source work.

Value of peer certification

Those with committer or PMC member status also get peer-certified for their work, making their recognized technical and social skills even more compelling and further reducing the hiring risk.

Rachel Chalmers of Ignition Partners, a boutique venture capital firm based in Silicon Valley, commented on the value of the work that open source software developers publicly perform:

When we look at a startup, we look at the GitHub repositories and Ohloh.net [now Open Hub]. We drill down to the level of individual developers. It informs our investment decision. That fact alone gives open source software developers significant leverage when negotiating their position, salary, and benefits with startups.

Value of positional power

A committer in an open source project of value to an employer has positional power that other job applicants

lack—the employer's products might be building on the project, for example. Regardless of the relevancy source, employers reap a number of benefits from hiring an applicant with open source experience:

- ▶ *Future visibility.* A committer is a project leader and thus has unique insight into the project's direction, which is strategically important to a company with products that build on the project or that offer a full foundation-managed platform.
- ▶ *Project influence.* The committer is in a position to channel the company's work into the project and to lead and inspire outside developers to contribute work that aligns with the employer's strategic goals.
- ▶ *Increased attractiveness to other prospective hires.* The committer's reputation attracts other developers, so by association the employer might be attractive to other competent developers, giving it a competitive edge.
- ▶ *Community goodwill.* The committer's community visibility is a positive reflection on that individual's employer. Paying the developer to work on the open source project creates community goodwill.
- ▶ *Competence by association.* Along with the developer's skills, the company acquires project competence. Trust in the company's products and services increases to the extent that they are related to the open source project, which helps the company's marketing and sales.

This increased insight, influence, and visibility are commercially pertinent to the company and hence afford the developer an improved negotiation position when it comes to salary and other job conditions.

Kai-Uwe Maetzel, an IBM employee in 2003 and later elected director of the Eclipse Foundation, observed:

Companies want to secure their influence on the Eclipse platform, and one way of doing so is by employing committers to Eclipse projects. Increasingly, I see regular developers being hired with the goal of making them committers within a few months.

THE INCREASED INSIGHT, INFLUENCE, AND VISIBILITY FROM OPEN SOURCE WORK ARE COMMERCIALLY RELEVANT AND STRENGTHEN THE NEGOTIATION POSITION.

When reminded of his remark for this article, Maetzel, who has left IBM to found to-do-list manager Task Krumplr, added:

My contributions to the Eclipse project resulted in high visibility in the Eclipse affine developer community. Pretty much every offer I received during these years from potential employers explicitly referred to my reputation in the Eclipse project.

Once an applicant is hired, the committer role is likely to provide higher job security as well. In hard times, the

economic value of the committer position might be the deciding factor in retaining the individual with that status.

Committers also have a richer job experience because the requirements are broader than those for a traditional developer. Their employers will expect them to perform well within the company, but might also expect them to keep working on the open source project. These expectations create a more rewarding work context and deepen committers' development experience.

LABOR MARKET INFLUENCES

Although contributors to an open source project are valuable, they are not scarce; many users become contributors and build a public reputation and

an open source resume. Committers are far fewer and thus enjoy more recognition and establish a reputation. At the high end of the ladder are PMC members and leaders. As a project matures and growth slows, it becomes more difficult to assume these more elite roles, simply because there are not that many positions to fill.

The labor market for developing open source components and building on them has few or no entry barriers. Open source software is readily available, and free educational materials abound. Anyone who wants to be part of the open source community can join

without running into financial or educational barriers.

Comments by Richard Seibt, president of the Open Source Business Foundation, support this free entry to the labor market:

With diminishing barriers to entry, everyone can contribute to open source and earn a living. Both software vendors and IT user firms are increasingly turning to open source foundations as a means for organizing software development. This provides ample employment opportunity for software developers who are skilled in open source development.

In contrast, closed source software development is exclusionary. To join this labor market, a developer must purchase a license and undergo commercial training. Sometimes, access is possible only through a mediator, such as an employer or academic institution. For the most part, no such complications exist for open source software development.

In addition, open source and associated Web services have facilitated the growth of end-user programmers—people without formal computer science education who can complete lightweight programming tasks using scripting languages and perform Web design. These individuals are also competing with traditional developers in the labor market.

As more software products build on open source components, the importance of these components will only increase. Consequently, well-heeled and highly skilled software developers may face more competition from

ABOUT THE AUTHOR

DIRK RIEHLE is the professor of open source software at Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany. His research interests include open source and inner source software engineering; agile software development methods; complexity science and human collaboration; and software system architecture, design, and implementation. Riehle received a PhD in computer science from ETH Zürich and an MBA from Stanford University's Graduate School of Business. He is a member of ACM, IEEE, and the German Informatics Society, and founder of the Open Symposium (OpenSym). Contact him at dirk@riehle.org or <http://dirkriehle.com>.

less wealthy and less skilled developers. Such competition was previously impossible due to labor market entry barriers in the then-dominant closed source software industry.

These trends suggest a class division in the software development labor market that will widen as more countries mature their development practices. Those who hold a powerful position in important open source projects will thrive, while those who remain in traditional, closed source practices will flounder amid increased global competition. Theoretically, this division will not be based on national origin, since the Internet affords nearly global access to open source software. Admittedly, however, Western developers have a head start: they created the ecosystem, defined its collaboration values, and have the spare time and resources to work on open source projects.

Open source project participation is creating a new career ladder with tangible benefits for software developers and employers. As more products incorporate open source software, the overall developer labor market is becoming more competitive. Closed source software that once afforded wealthy Western developers some protection is giving way to open source projects that are eroding the expertise silo. In the future, people from low-wage countries that are not already in the developer market will join, intensifying competition. Developers with open source project experience will have the edge over those still relying on their closed source software development skills. ■

ACKNOWLEDGMENTS

I thank Ann Barcomb, Hannes Dohrn, Kai-Uwe Maetzel, Michael Maximilien, and Robert O'Callahan for helping me improve this article. I also thank Chris DiBona, Justin Erenkrantz, Kai-Uwe Maetzel, Marten Mickos, Robert O'Callahan and Richard Seibt for their timely provision of quotes that illustrate the impact of open source on the software developer career.

REFERENCES

1. D. Riehle et al., "Paid vs. Volunteer Work in Open Source," *Proc. 47th IEEE Hawaii Int'l Conf. System Science (HICSS 14)*, 2014, pp. 3286–3295.
2. D. Riehle, "The Economic Motivation of Open Source: Stakeholder Perspectives," *Computer*, vol. 40, no. 4, 2007, pp. 25–32.
3. N. Kimmelman, "A Career in Open Source? Relevant Competencies for Open Source Software Developers," *Information Technology*, vol. 55, no. 5, 2014, pp. 204–211.
4. K. Crowston and J. Howison, "The Social Structure of Free and Open Source Software Development," *First Monday*, vol. 10, no. 2, 2011; <http://firstmonday.org/article/view/1207/1127>.
5. A. Binstock, "How to Contribute to Open Source Projects," *Dr. Dobbs's*, 21 June 2011; <http://drdobbs.com/open-source/231000080>.
6. C. Jensen and W. Scacchi, "Role Migration and Advancement Processes in OSSD Projects: A Comparative Case Study," *Proc. 29th IEEE Int'l Conf.*

Software Eng. (ICSE 07), 2007, pp. 364–375.

7. P. Weißgerber et al., "Small Patches Get In!," *Proc. IEEE Int'l Working Conf. Mining Software Repositories (MSR 08)*, 2008, pp. 67–76.
8. D. Riehle, "The Economic Case for Open Source Foundations," *Computer*, vol. 43, no. 1, 2010, pp. 86–90.
9. J. Lerner and J. Tirole, "Some Simple Economics of Open Source," *J. Industrial Economics*, vol. 50, no. 2, 2002, pp. 197–234.
10. D. Riehle et al., *Inner Source in Platform-Based Product Engineering*, tech. report CS-2015-02, Dept. of Computer Science, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany, 2015.
11. I.H. Hann, "Why Do Developers Contribute to Open Source Projects? First Evidence of Economic Incentives," *Proc. 2nd IEEE Workshop Open Source Software Eng.*, 2002, <http://flosshub.org/system/files/HannRobertsSlaughterFielding.pdf>
12. J. Bitzer et al., *Returns to Open Source Engagement: An Empirical Test of the Signaling Hypothesis*, working paper V-321-10, Univ. of Oldenburg, Germany, 2010.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.

