



A New Developer Career

Prof. Dr. Dirk Riehle, M.B.A.

Friedrich-Alexander-University of Erlangen-Nürnberg

Linux-Tag 2010: June 10, 2010

This document is licensed under the Creative Commons BY-SA 3.0 license.
© Copyright 2010 Dirk Riehle, see <http://dirkriehle.com>. Some Rights Reserved.

Community Open Source

Software Developer Careers

Economic Value of Participation

Becoming a Committer

Conclusions



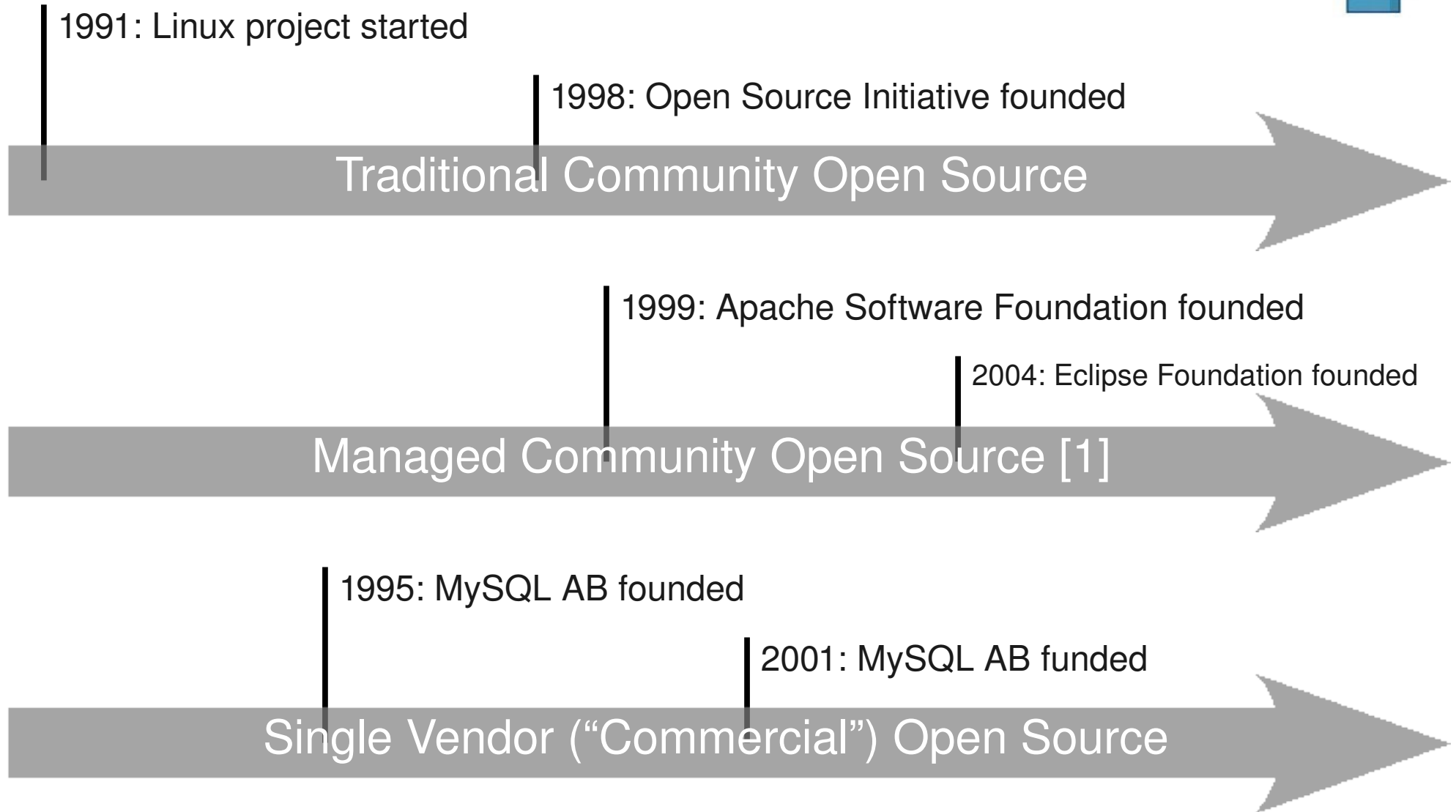
PART I

Community Open Source

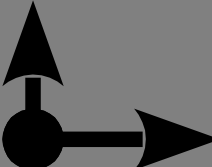
		Project Type	
		Single product or product line	Multi-product assembly
Ownership	Community-owned	Community Open Source (e.g. Linux, Apache)	Community Distribution (e.g. Debian)
	Single owner or dominant vendor	Single-Vendor Open Source (e.g. MySQL, Alfresco)	Single-Vendor Distribution (e.g. RHEL, SLES)

[1] Dirk Riehle. "The Economic Motivation of Open Source: Stakeholder Perspectives." IEEE Computer, vol. 40, no. 4 (April 2007). Page 25-32.

Community Open Source	Single-Vendor Open Source
Community of owners	Single proprietor
Single license	Multiple licenses
No functionality withheld	Feature differentiated
Cross-subsidized	Venture-capital backed
Minimal direct revenues	Significant direct revenues
Community of equals	Asymmetric community



[1] Dirk Riehle. "The Economic Case for Open Source Foundations." IEEE Computer, vol. 43, no. 1 (January 2010). Page 86-90.

		Openness	
		Open development	Closed development
Ownership	Community-owned	Open Source Foundations (e.g. Apache, Gnome)	Open Source Consortia (e.g. OW2, GenIVI)
	Single owner or dominant vendor		Single-Vendor Open Source (e.g. MySQL, Alfresco)

Egalitarian

Meritocratic

Self-Organizing

[1] Dirk Riehle, John Ellenberger, Tamir Menahem, Boris Mikhailovski, Yuri Natchetoi, Barak Naveh, Thomas Odenwald. "Open Collaboration within Corporations Using Software Forges." IEEE Software, vol. 26, no. 2 (March/April 2009). Page 52-58.

Open Collaboration

- Egalitarian
 - Open for contribution
 - Everyone can contribute
- Meritocratic
 - Public discussion process
 - Decisions based on merit
- Self-organizing
 - People find their own process
 - People find their best project

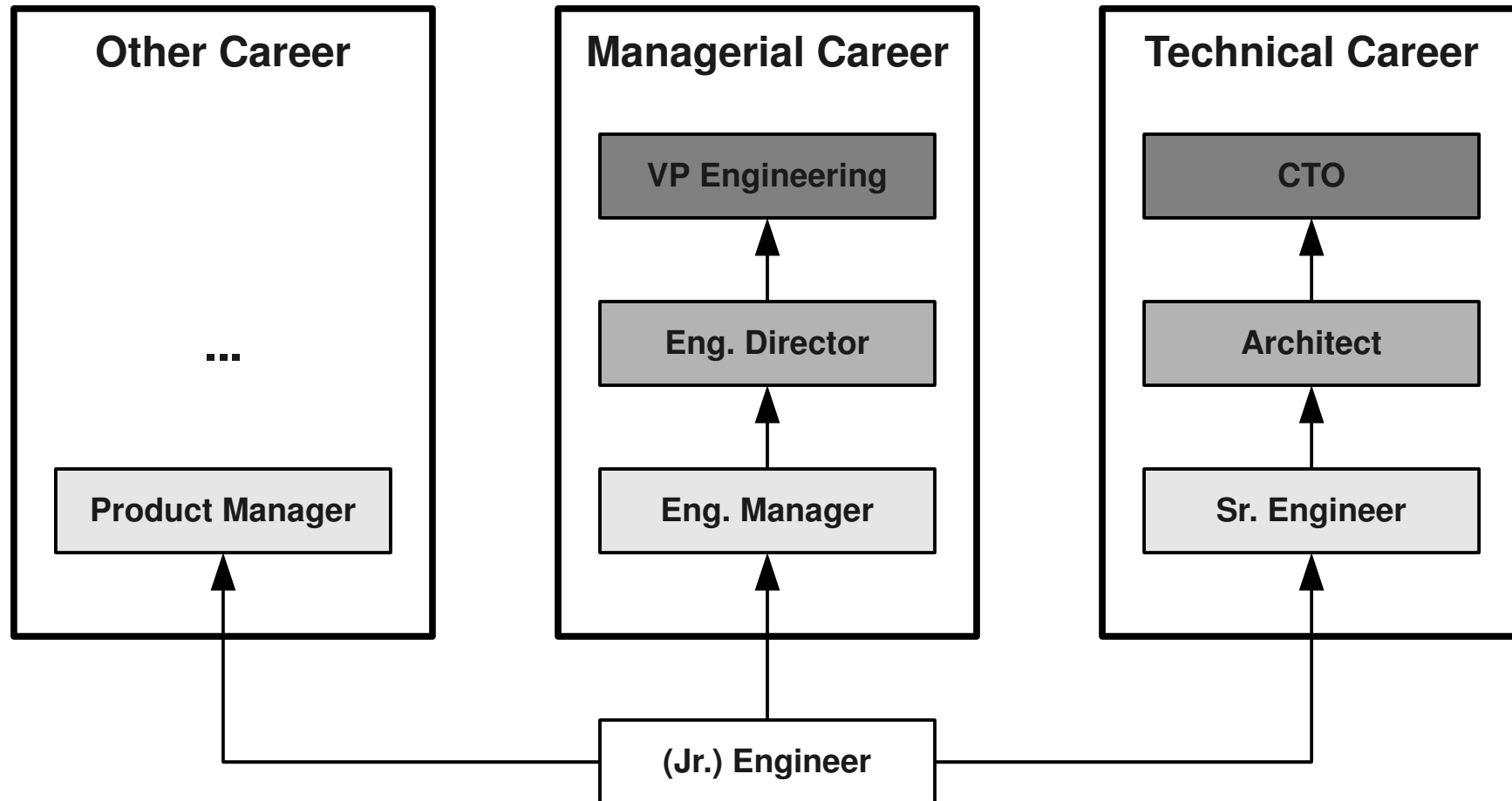
Traditional Work

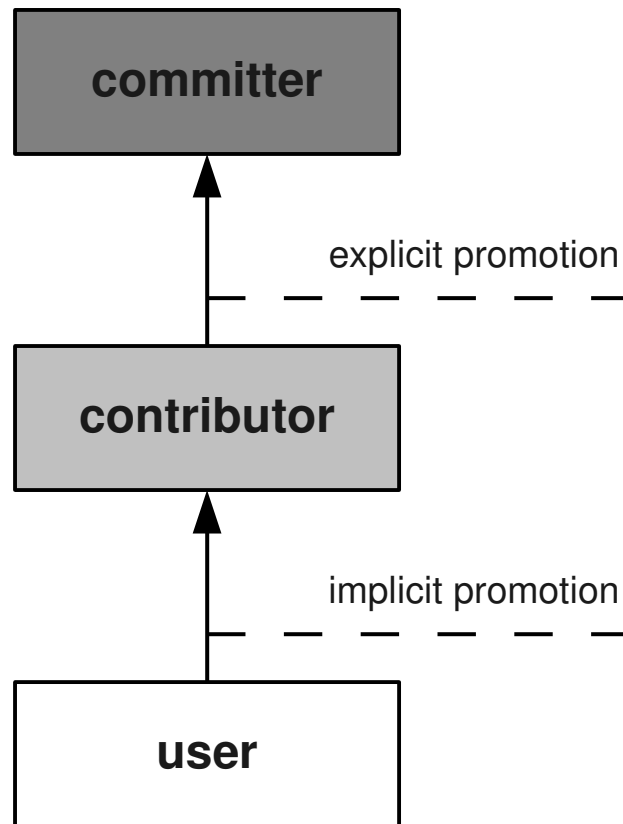
- Hierarchical
 - Closed and hidden silos
 - Assigned to project
- Status-oriented
 - Public and private discussions
 - Hierarchical status decides
- Assigned tasks
 - Prescribed process
 - Prescribed jobs



PART II

Software Developer Careers





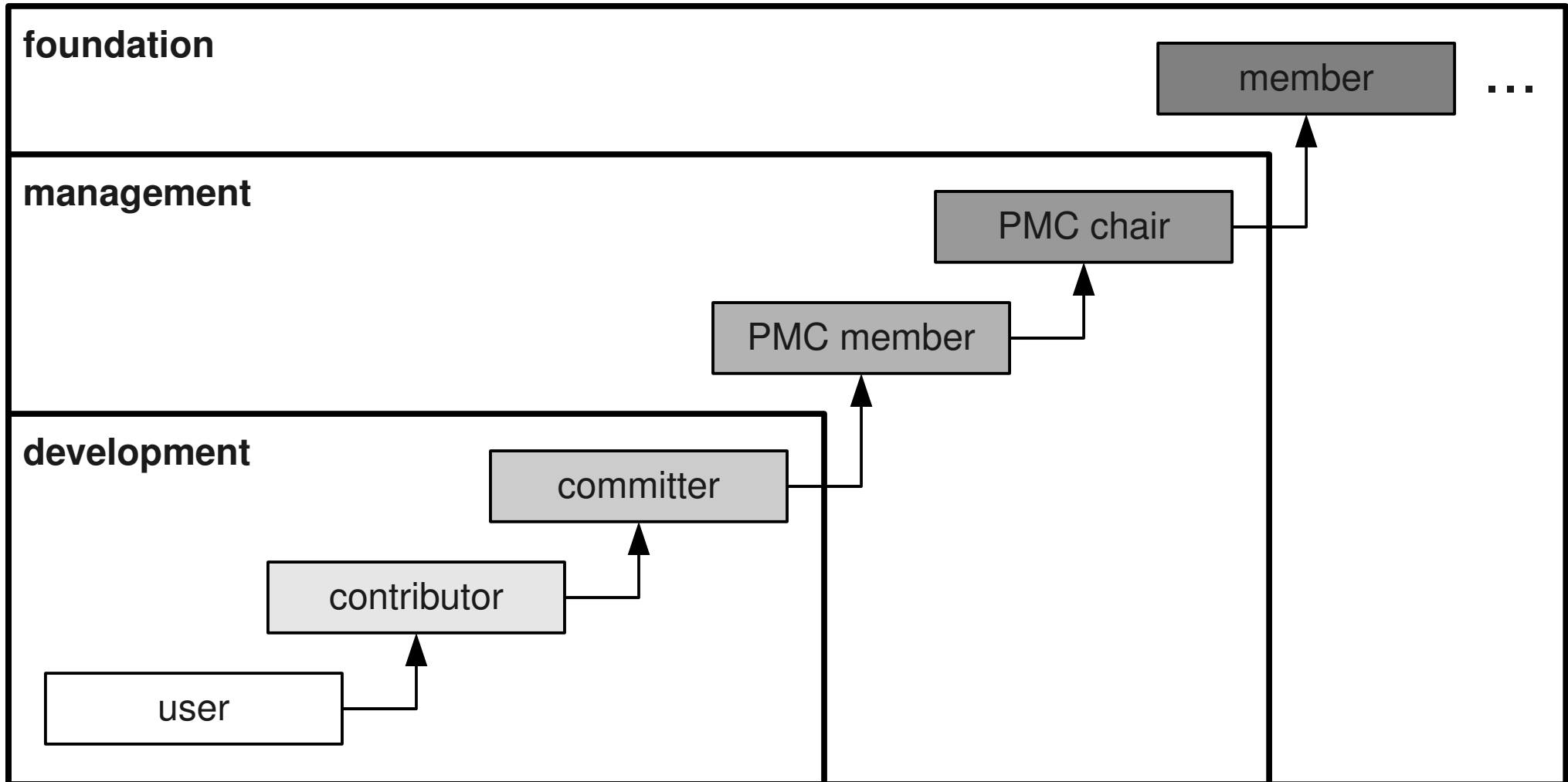
- Formally: Has commit (write) rights
 - Performs bulk of work; quality assurance
-
- Provides small features, bug fixes
 - Submits patches (no commit rights)
-
- Knows and uses software
 - If at all, helps with comments, feedback

In community open source, and
under an open development model,

committer status is earned

while in single-vendor open source, or
under a closed development model,

committer status is assigned





PART III

Economic Value of Participation

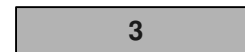
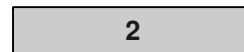
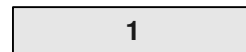
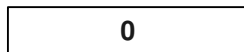
Project Skills

Peer Certification

Leadership Position

		Project Member Status		
		user	contributor	committer
Project Relevance	low / none		Peer Certification	Peer Certification
	high / some	Project Skills	Peer Certification Project Skills	Project Skills Peer Certification Leadership Role

Legend



What Position Affords	Value to Employer
Validated technical abilities	Reduced hiring risk
Deeper insight, more leverage	Better work product quality
Community visibility, reputation	Higher reputation, more sales
Strategic alignment with project	Lower costs, more predictability

- Successful open source contribution implies technical skills
- Skills are “peer-certified” and publicly documented (rather than hidden)
- Validated technical abilities (peer certification) reduce hiring risk
- Reduced hiring risk lowers uncertainty discount in wage negotiations

- Leadership position first implies deeper insight, higher leverage
 - A committer has a deeper understand and insight into the project
 - A committer knows who to talk to and which knobs to turn
- Deeper insight, higher leverage implies better work quality
 - Problems get identified and fixed faster

- Leadership position first implies higher reputation with user community
 - Higher reputation and visibility make leader a go-to-person wrt project
 - Higher likelihood for talks, panels, articles follows
- Higher reputation of leader transfers to employer
 - Higher reputation of employer implies higher work quality wrt project
 - Higher reputation and expected work quality imply more sales
 - Higher reputation imply fee premium for services

- Leadership position first implies better alignment with project
 - Inside view allows for faster recognition of what's likely to happen
 - Leadership position helps form future strategic direction of project
- Better strategic alignment improves predictability, lowers costs
 - Being aligned earlier, better, removes uncertainty and waste, thus lowers costs
 - Being aligned sets up employer to utilize new opportunities faster than others

Higher Salary

More Job Security

Higher Job Versatility

Richer Job Experience

- Lerner and Tirole's “signaling hypothesis” [1]
 - Developers offset opportunity costs by signaling marketable skills to future employers
 - Open source performance is public, more precisely measurable, and less captive
 - At the publication's time (2002), authors say volunteer developers were dominant
- Hann et al.'s 2000 empirical investigation [2]
 - Open source contributions in themselves don't correlate with higher wages
 - Higher status within merit-based Apache Software Foundation project does
 - » Rank “committer” or higher showed 29% higher salary after controlling for most variables
 - » Study indicates that committer status is a proxy for otherwise unobservable skills
- Bitzer et al.'s 2006/07 “c't wage survey” empirical investigation [3]
 - 61% of respondents believe open source activity benefits their career
 - Study shows no wage premium for voluntary open source contributions

[1] Lerner, J. and J. Tirole (2002). “Some Simple Economics of Open Source.” Journal of Industrial Economics, vol. 50, no. 2, pp. 197-234.

[2] Hann, I-H. et al. (2002). “Why Do Developers Contribute to Open Source Projects?” 2nd OSS Workshop, 2002.

[3] Bitzer, J. et al. (2010). “Returns to Open Source Engagement: An Empirical Test of the Signaling Hypothesis.” Univ. Oldenburg, 2010.

- Developers view open source contributions as beneficiary
- Explicit signaling of committer status on social networks
- Explicit inclusion on resume and with links to contributions
- Informal discussions with hiring managers indicating preference
- Increased hiring for open source work on company time
- Hiring with set performance goal to become committer in defined time
- Hiring specific committers with added loyalty incentives



PART IV

Becoming a Committer

- Need technical competence on project
 - Typically programming skills, but real projects need more
 - Configuration, administration, documentation, website, and ...
 - An understanding of the processes of how systems are built
- Social skills are equally important if not more so
 - Need communication and collaboration skills for team work
 - Need good reading and writing skills (email communication)
 - Leadership skills are always rare and in demand
 - Need to understand peer behavior, subculture
- All things being equal, social skills are more important

- Achieving committer status is a communal process
 - Users work their way up from user through contributor to committer status
 - This creates acquaintance, credibility, reduces risk of wrong decision
 - Eventually, contributor is put up for committer status
 - Decision is typically made by vote of existing committers
- There is good and there is bad timing
 - A growing project needs more people, will give more responsibility early on
 - A mature project with slowing growth rarely needs more committers
 - Try to get in early when every helping hand is needed
- The prospective committer's dilemma: Which project to bet on?



PART V

Conclusions

- A typical asymmetric market (less employers than employees)
 - Traditionally, work remains hidden (cf. reduced hiring risk)
- Contributor status does not provide a sustainable advantage
 - Everyone who is smart enough can contribute and gain recognition
 - No barriers to entry, there is always one more feature to implement
 - No barriers, because open source is freely available to everyone
- Committer status does provide sustainable positional advantage
 - Status has economic value, is not achieved or handed out easily
 - Mature projects are largely closed to new committers
 - Once achieved, committer status is not taken away easily

What Position Affords	Value to Employer
Validated technical abilities	Reduced hiring risk
Deeper insight, more leverage	Better product quality
Community visibility, reputation	Higher reputation, more sales
Strategic alignment with project	Lower costs, more predictability

Higher Salary

More Job Security

Higher Job Versatility

Richer Job Experience

- Regular developers
 - Employed to develop proprietary custom code
 - Know and use open source, may or may not contribute
 - Not having contributed to any open source project becomes a stigma
 - Lower barriers to labor market entry push down salaries
- Open source committers
 - Employed to work (full-time) on commercially relevant open source
 - Committers' economic value (and loyalty?) is with open source project
 - Committers likely to behave more like free agents than employees
 - Scarcity of committer status may provide superior compensation
- Developer labor market may become a two-class society

- New career path emerged; is related but different to prior paths
- In the short-run, open source developers may improve their wages
- In the long-run, some form of peer-certification becomes a must
- Reduced barriers to market entry make life harder for developers
- The way out is to become a committer, but that is a scarce status
- Committers to relevant open source projects will benefit well

This document is licensed under the Creative Commons license BY-SA 3.0, see <http://creativecommons.org/licenses/by-sa/3.0/>

The document's original author and copyright holder is Dirk Riehle. The author can be reached at dirk@riehle.org. The document can be found at <http://dirkriehle.com>.

You are allowed to share and create derivatives of this document, even for commercial purposes, but you must keep this attribution page included in the document and you must show it as part of your performance. The title slide must visibly bear the following two lines of text:

This document is licensed under the Creative Commons BY-SA 3.0 license.
© Copyright 2010 Dirk Riehle, see <http://dirkriehle.com>. Some Rights Reserved.

You may add your own copyright notice for the portions of the document you added.



Questions? Feedback!

<http://osr.cs.fau.de> - dirk.riehle@cs.fau.de - [@osrgroup](#)

<http://dirkriehle.com> - dirk@riehle.org - [@dirkriehle](#)

Thank you!