

# Open Source Software Research: Communities, Companies and the Future

Puzzle ITC, [www.puzzle.ch](http://www.puzzle.ch)  
15. März 2006, Bern

Matthias Stürmer  
[www.stuermer.ch](http://www.stuermer.ch)

# Übersicht

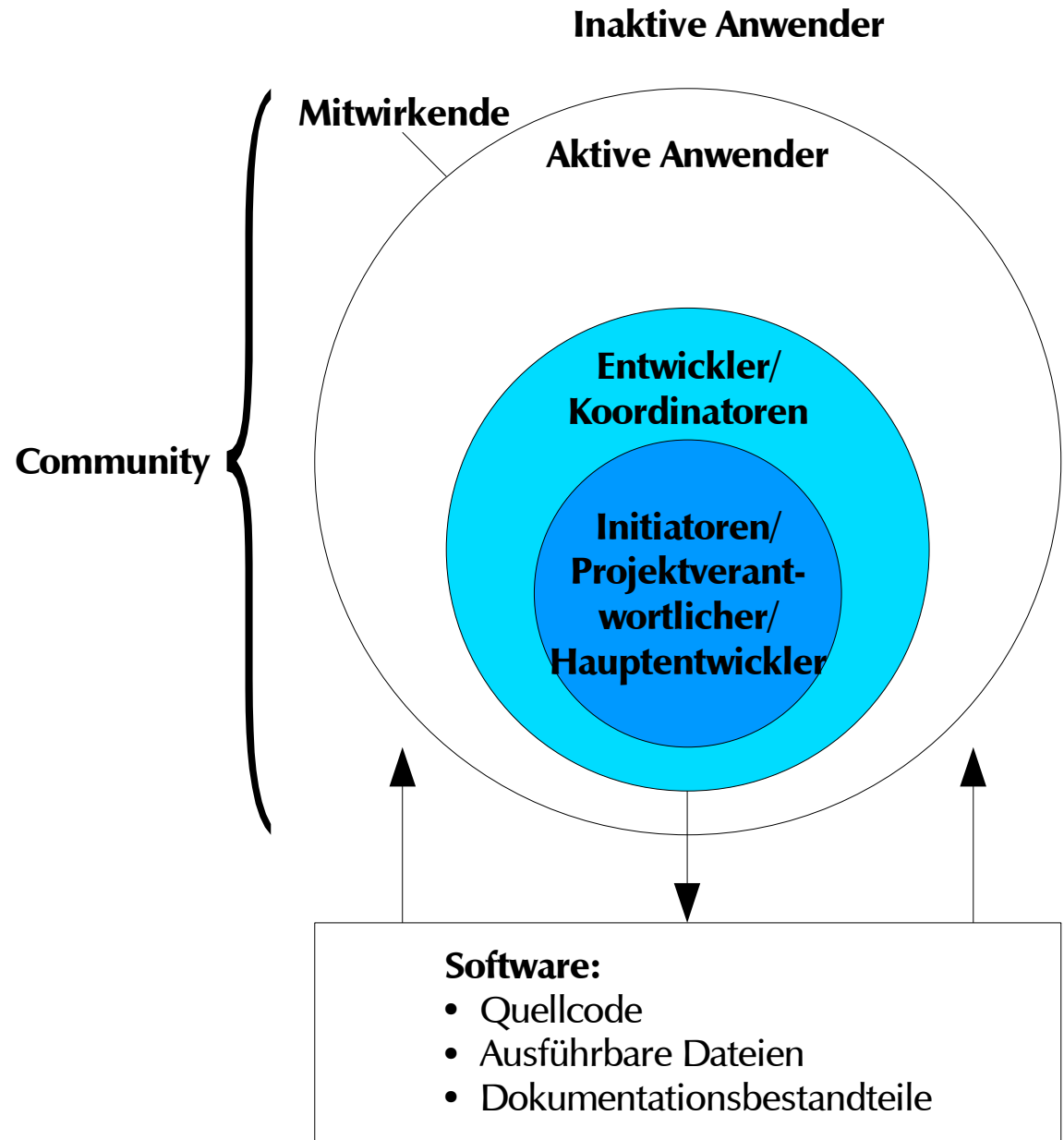
- **Teil A: Open Source Community Building**
- Teil B: Engagement von Firmen in Open Source Projekten
- Teil C: Open Source Software Forschung an der ETH Zürich

# Untersuchte Open Source Projekte



- Bernhard Bühlmann
- Boris Kraft
- Bertrand Delacrétaz
- Guido Wesdorp
- Michael Wechner
- Daniel Hinderink
- Bard Farstad
- Gregor Rothfuss

# Schema einer Open Source Community



# Charakteristiken von erwünschten Communities

- 1. Produktivität:** keine endlosen Diskussionen, keine Trittbrettfahrer
- 2. Selbstmotivation:** Ideen von Anfang bis Ende selber realisieren
- 3. Vielfältigkeit:** Unabhängigkeit von lokalen Veränderungen
- 4. Korrektheit:** Respektvoller Umgang, keine giftigen Antworten
- 5. Altruismus:** Keine One-Man-Shows, fruchtbare Erde bewahren
- 6. Ausdauer:** Langfristiges Engagement ist wertvoll
- 7. Vision:** Gemeinsam in die gleiche Richtung gehen

# Eigenschaften und Verhalten von Projektkoordinatoren

- Bestimmtheit
- Engagement
- Erfahrung
- Geduld
- Hilfsbereitschaft
- Offenheit
- Mitteilsamkeit
- Persönlichkeit
- Präsenz
- Programmiertalent
- Verantwortungsbereitschaft
- Visionär

**Fazit: Einer alleine kann's nicht schaffen,  
es braucht mehrere Koordinierende.**

# Ausgangslage eines Open Source Projekts

- 1. Programmiersprache:** Nur "theoretische" Auswahl, Einfluss auf Pool potentieller Programmierer
- 2. Open Source Lizenz:** Viral/Non-viral, Einfluss auf Community-Motivation und -Bildung
- 3. Ausgangs-Quellcode:** Potential der Software ersichtlich
- 4. Nachfrage:** Zeittor für günstigen Moment erwischen
- 5. Innovationsgrad:** Von Kopie bis radikal neu
- 6. Anwendbarkeit:** Z.B. Durch Plattformunabhängigkeit
- 7. Kommunikation:** Hohe Anfangsinvestitionen nötig

# Förderung des Community Building

- Ebenen: **Rekrutierung; Zusammenarbeit; Produktivität**
- Gebiete:
  - **Modularität:** Software soll modular programmiert werden
  - **Dokumentation:** Verschiedene Interessensgruppen berücksichtigen
  - **Release Management:** Release Management Prozess einführen
  - **Kollaborationsplattform:** Eine Umgebung der Zusammenarbeit verwenden
  - **Physische Begegnungen:** Physische Treffen veranstalten
  - **Trägerorganisation:** Organisatorische Trägerschaft des Projekts gründen
  - **Internationalisierung:** Eine internationale Community anstreben

# Empfehlungen pro Gebiet und Ebene

↓ Gebiet	Ebene →	<b>Rekrutierung</b>	<b>Zusammenarbeit</b>	<b>Produktivität</b>
<b>Modularität</b>		Beschreibung zur Entwicklung von Erweiterungen zur Verfügung stellen	Bei vielen Erweiterungen Qualitätssicherung durch erfahrene Programmierer einführen	Erweiterbarkeit der Software durch externe Komponenten von Anfang an vorsehen
<b>Dokumentation</b>		Wenige dafür aktuelle und klar strukturierte Tutorials und Handbücher verfügbar machen	Anreizsystem für die Erstellung von hochwertiger Dokumentation schaffen	Erläuterungen im Quellcode und zum Application Programming Interface schreiben
<b>Release Management</b>		Regelmässig häufig Software-Releases inklusive aussagekräftiger Ankündigungen herausgeben	Fairer Veröffentlichungsprozess, u.a. mit Feature Freeze, praktizieren	Rückwärtskompatibilität gewährleisten und wenn nicht möglich Migrationskripte entwickeln
<b>Kollaborationsplattformen</b>		Projekt auf den bekannten Plattformen unter kennzeichnenden Stichwörtern registrieren	Vorhandene bzw. individuelle Entwicklungs- und Zusammenarbeits-Plattform aufbauen	Den Entwicklern ein hochverfügbares Revision Control System zur Verfügung stellen
<b>Physische Begegnungen</b>		Präsentationen und Workshops an Konferenzen und Messeveranstaltungen geben	Community-Treffen zur technischen und organisatorischen Zusammenarbeit organisieren	Entwickler-Sprints in motivierender Umgebung und mit klaren Zielen veranstalten
<b>Trägerorganisation</b>		Das Projekt einer bestehenden Trägerschaft anschliessen oder eine neue gründen	Organisatorische, rechtliche und repräsentative Aufgaben durch die Trägerschaft wahrnehmen	Entwicklertätigkeiten durch die Trägerschaft unterstützen und beschützen
<b>Internationalisierung</b>		Auf Englisch kommunizieren und Übersetzungs-Aufgaben ausschreiben	In einer multikulturellen Community höflich und respektvoll miteinander kommunizieren	Mehrsprachigkeit der Software auf technischer Ebene von Anfang an vorsehen

# Ebenen-spezifische Tätigkeiten

↓ Gebiet	Ebene →	Rekrutierung	Zusammenarbeit	Produktivität
<b>Bekanntmachung</b>		Herkömmliche Marketingaktivitäten betreiben und Pressemitteilungen verfassen		
<b>Credit System</b>		Alle Beiträge mit deutlichem Hinweis auf Autoren veröffentlichen		
<b>Kommunikationskanäle</b>			Entsprechend der Community-Grösse Kommunikation auf einige wenige Kanäle fokussieren	
<b>Community-Struktur</b>			Nur den Bedürfnissen der Community entsprechend Strukturen schaffen	
<b>Aufgabenliste</b>			Kurze Beschreibungen anstehender Aufgaben mit Aufforderung zur Mithilfe publizieren	
<b>Software-Qualität</b>				Nur funktionierende und ausgereifte Software-Änderungen und -Erweiterungen integrieren
<b>Benutzeroberfläche</b>				Benutzeroberfläche für bessere Bedienbarkeit und Gestaltung ausarbeiten
<b>Installation</b>				Installationsprozess anhand eingeholter Feedbacks optimieren

# Do's und Don'ts von Projekt-Koordinatoren

- Tue es für dich.
- Verliere dich nicht in Perfektionismus.
- Akzeptiere Beiträge von anderen.
- Kommuniziere offen.
- Schlage nicht Ideen vor sondern trage zur Lösung bei.
- Verhalte dich freundlich.
- Betreibe intensives Marketing.

**Frage: Was sind eure Erfahrungen, Do's und Don'ts?**

# Übersicht

- Teil A: Open Source Community Building
- **Teil B: Engagement von Firmen in Open Source Projekten**
- Teil C: Open Source Software Forschung an der ETH Zürich

# Von Firmen gegründete OS-Projekte

- **Vorteile:**

- Kontinuität in der Weiterentwicklung
- Angebot professioneller Dienstleistungen
- Verantwortungsbewusstsein durch Kundenbindungen (z.B. bezüglich Rückwärtskompatibilität)

- **Herausforderungen:**

- Eingegrenzte Mitwirkungsmöglichkeiten Aussenstehender
- Lizenz-Problematik wenn Unternehmen Copyright behalten wollen
- Kritische Beziehung zur Community: Geld verdienen, Firmenpräsenz etc.
- Kommunikation muss öffentlich werden (z.B. keine Firmen-internen Gespräche mehr, sondern nur noch über Mailing List kommunizieren)

# Motivation für Firmen, sich in OS Projekten zu engagieren

## 1. Strategische Gründe:

- Unabhängigkeit von der Preis- und Lizenzpolitik grosser Software-Unternehmen
- Erhalten von Innovationen und Entwicklungsleistungen aus der Community
- Hervorragende Entwickler als zukünftige Mitarbeitende finden
- Konkurrenz schwächen durch Veröffentlichen von Software

# Motivation für Firmen, sich in OS Projekten zu engagieren

## 2. Wirtschaftliche Gründe:

- Verkauf von Dienstleistungen für die Software
- Verkauf von Software, welche dem OS Projekt nahe steht
- Marketing-Kosten senken durch Gratis-Marketing dank populärem OS Projekt
  - Demonstration von technischem Können
  - Verbreitung des Firmennamens (z.B. Puzzle-Logo dank SourceForge Mirror ;)
  - Stärkung der Partner-Position
  - Profitieren, wenn das OS Projekt bei populären Open Source Brands wie Apache, Mozilla etc. publiziert ist

# Motivation für Firmen, sich in OS Projekten zu engagieren

## 3. Technische Gründe:

- Weiterentwicklungen von den Entwicklern, besonders:
  - Extension-Entwicklung
  - Betriebssystem-Unabhängigkeit
  - Hardware-Unabhängigkeit
- Dokumentation, Testing, Bugreporting von den Benutzern
- Sicherheit und Zuverlässigkeit der Software verbessern

# Motivation für Firmen, sich in OS Projekten zu engagieren

## 4. Soziale Gründe:

- Den Werten der Community anpassen, um unentgeltlich arbeitende Entwickler nicht zu verärgern
- Source Code und Wissen mit der Community teilen um deren Unterstützung nicht zu verlieren
- Grundüberzeugung vertreten, Software sollte nicht verschlossen sein, um Mentalität von gewissen Software-Konzernen zu widersprechen

# Viele Gründe, sich als Firma in Open Source Projekten zu engagieren

1. Strategische Gründe
2. Wirtschaftliche Gründe
3. Technische Gründe
4. Soziale Gründe

**Frage: Welches sind Nachteile, wenn Firmen Open Source Software entwickeln?**

# Unterschiedliche Geschäftsmodelle

## 1. Dienstleistungen anbieten

Community-initiierte Projekte als Ursprung für Bedürfnis nach Unternehmensleistungen: Beratung, Installation, Verteilung, Anpassungen, Schulungen, Wartung, Weiterentwicklung

## 2. Entwicklung von Erweiterungen

Basis-Software ist unter OSS-Lizenz, Erweiterung sind proprietär

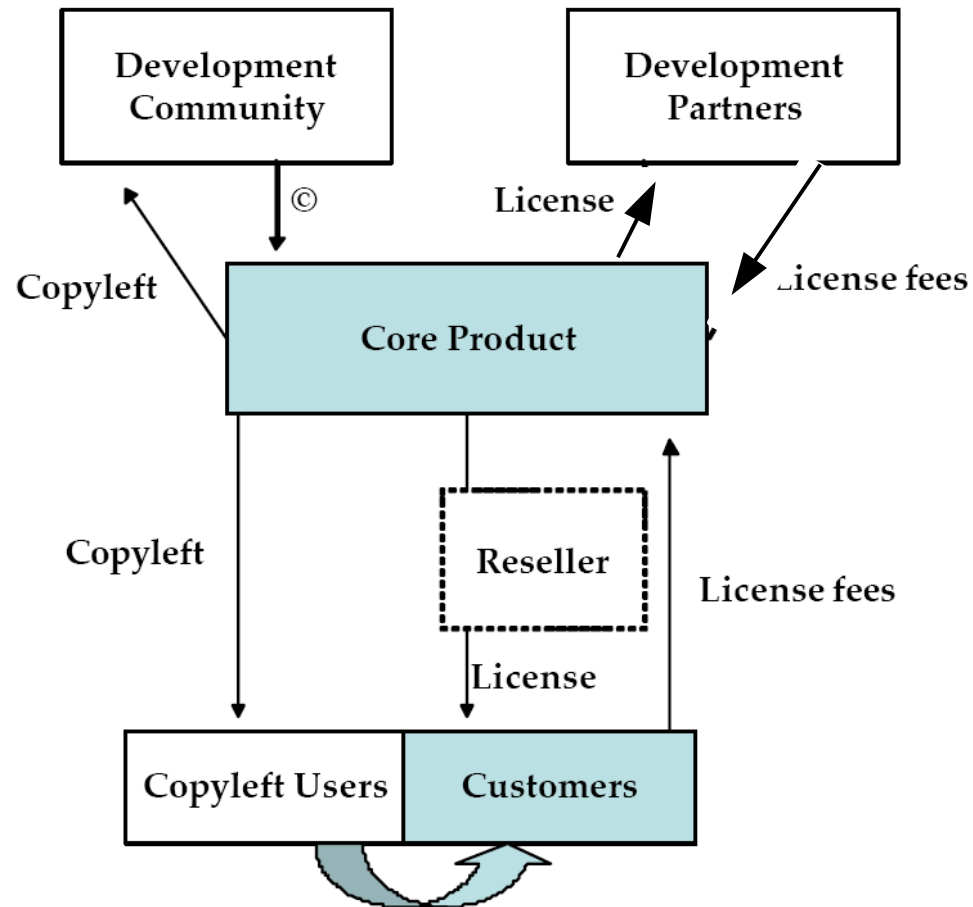
## 3. Partnerschafts-Programm

Kostenpflichtige Partnerschaften mit anderen Software-Herstellern

## 4. Dual Licensing

GPL-Software an Community, proprietäre Lizenz an zahlende Kunden, externen Entwickler müssen Copyright abgeben

# Das Prinzip des Dual Licensing



# Übersicht

- Teil A: Open Source Community Building
- Teil B: Engagement von Firmen in Open Source Projekten
- **Teil C: Open Source Software Forschung an der ETH Zürich**

# Auszug bisheriger OSS Forschungsprojekte am Lehrstuhl von Krogh

- **2002: "Private-Collective" Innovation Model:**  
Weiterentwicklung der Innovations-Theorie: Entstehung von OSS kombiniert zwei bisher unabhängige Innovations-Modelle, "Private Investment" und "Collective Action"
- **2003: Community, Joining and Specialization:**  
Wie funktionieren Communities intern? Wie wird man Mitglied? Wie spezialisieren sich die Mitglieder?
- **2006: Knowledge Reuse in Open Source Software:**  
Warum und wann werden Software-Komponenten in Open Source Projekten häufig wiederverwendet?

# Aktuelles Forschungsprojekt

## Design for Reuse

- *Forschungsfrage*: Weshalb gibt es wiederverwendbare Open Source Software, wo es doch bis zu 200% Mehraufwand bedeutet Software wiederverwendbar zu entwickeln?
- *Methode*: Interviews mit Entwicklern von Komponenten, welche in Open Source Projekten wiederverwendet werden
- *Bereits untersuchte Komponenten*: zlib, libpng, expat, libxml, nanoxml, adodb, pango, libgcrypt
- *Bisherige Motive*: Technisches Problem generisch lösen, Standard pushen, Vorteile von breiter Verwendung, Reputations-Steigerung, verbessern der Job-Chancen, idealistische Gründe

# Mögliche zukünftige Forschungsthemen

## Corporate Community Building

- Wann macht's Sinn für Unternehmen Software unter OS-Lizenzen zu veröffentlichen, wann nicht? Generell: Wann sollen Unternehmen ihre Innovationen teilen? (Thema Open Innovation)
- Wie können Unternehmen mit Communities optimal zusammenarbeiten? Wie können sie Einfluss auf die Weiterentwicklung der Software ausüben?
- Welche Arbeiten können Unternehmen effizient ausführen, bei welchen sind Communities erfolgreicher?
- Einfluss von Bounty-Systemen (Google Summer of Code)?

# Mögliche zukünftige Forschungsthemen

## Open Source Success

- Wie kann der Erfolg von Open Source Projekten definiert werden? Welche relevanten Interessensgruppen gibt es?
- Welches sind die Faktoren, die ein erfolgreiches Projekt auszeichnen? Welche Eigenschaften lassen ein Projekt sterben?
- Welches sind erfolgreiche Geschäftsmodelle, welche haben sich nicht bewährt?
- Welchen Einfluss hat OSS auf die weltweite Software Branche?

# Fragen aus der Praxis?

- Was sind eure Fragen bezüglich Open Source Software?
  - Über welche Themen sollte mehr publiziert werden?
  - Welche Software-Trends erlebt ihr im Alltag?
- **Meine Meinung: Forschung muss schliesslich der Praxis dienen!**

*Merci für eure Aufmerksamkeit!*

# Quellenangaben

- Bonaccorsi, A. und C. Rossi 2003: 'Comparing motivations of individual programmers and firms to take part in the Open Source movement. From community to business', Sant' Anna School of Advanced Studies, Italy.  
<http://opensource.mit.edu/papers/bnaccorsirossimotivationlong.pdf>
- Häfliger, Stefan, Georg von Krogh, Sebastian Späth 2006: 'Knowledge Reuse in Open Source Software', Revised Version for Management Science.
- Stürmer, Matthias 2005: 'Open Source Community Building', Lizenziatsarbeit Universität Bern.  
[http://stuermer.ch/blog/documents/Stuermer\\_2005\\_OpenSourceCommunityBuilding.pdf](http://stuermer.ch/blog/documents/Stuermer_2005_OpenSourceCommunityBuilding.pdf)
- Stürmer, Matthias und Thomas Myrach 2006: 'Open Source Community Building', *Open Source Jahrbuch 2006*.  
[http://stuermer.ch/blog/documents/StuermerMyrach\\_2006\\_OpenSourceCommunityBuilding.pdf](http://stuermer.ch/blog/documents/StuermerMyrach_2006_OpenSourceCommunityBuilding.pdf)
- Stürmer, Matthias 2005: 'Business Opportunities of Companies Active in Open Source Projects', Sonderstudie Universität Bern.  
[http://stuermer.ch/blog/documents/Stuermer\\_2005\\_BusinessOpportunitiesOfCompaniesActiveInOpenSourceProjects.pdf](http://stuermer.ch/blog/documents/Stuermer_2005_BusinessOpportunitiesOfCompaniesActiveInOpenSourceProjects.pdf)
- Välimäki, M. 2003: 'Dual Licensing in Open Source Software Industry', *Systemes d'Information et Management* 1/2003.  
<http://opensource.mit.edu/papers/valimaki.pdf>
- von Hippel, Eric und Georg von Krogh 2003: 'Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science', *Organization Science* 14/2: 209-223.
- von Krogh, Georg, Sebastian Späth und Karim R. Lakhani 2003: 'Community, joining, and specialization in open source software innovation: a case study', *Research Policy* 32: 1217-1241. <http://opensource.mit.edu/papers/rp-vonkroghspaethlakhani.pdf>
- West, Joel und Siobhan O'Mahony 2005: 'Contrasting Community Building in Sponsored and Community Founded Open Source Projects', *Proceedings of the 38th Annual Hawai'i International Conference on System Sciences* 2005.