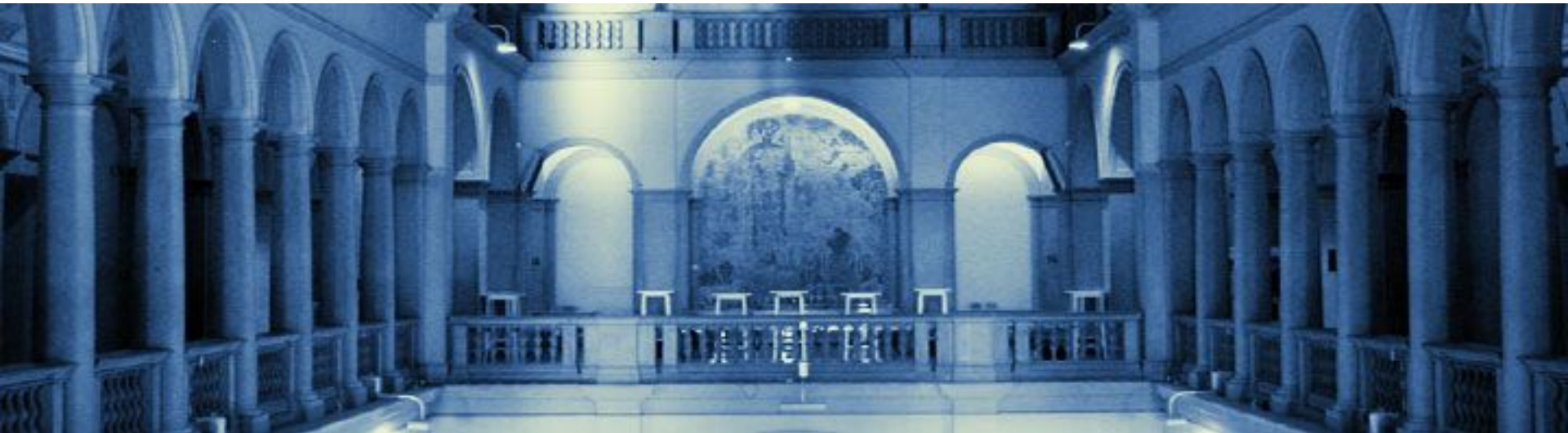


Open Source Community Building

Perspektiven auf Makro- und Mikro-Ebene

6. Juni 2007 – SIK–Arbeitsgruppentagung OSS – Bern
Matthias Stürmer, ETH Zürich & /ch/open – mstuermer@ethz.ch



Übersicht

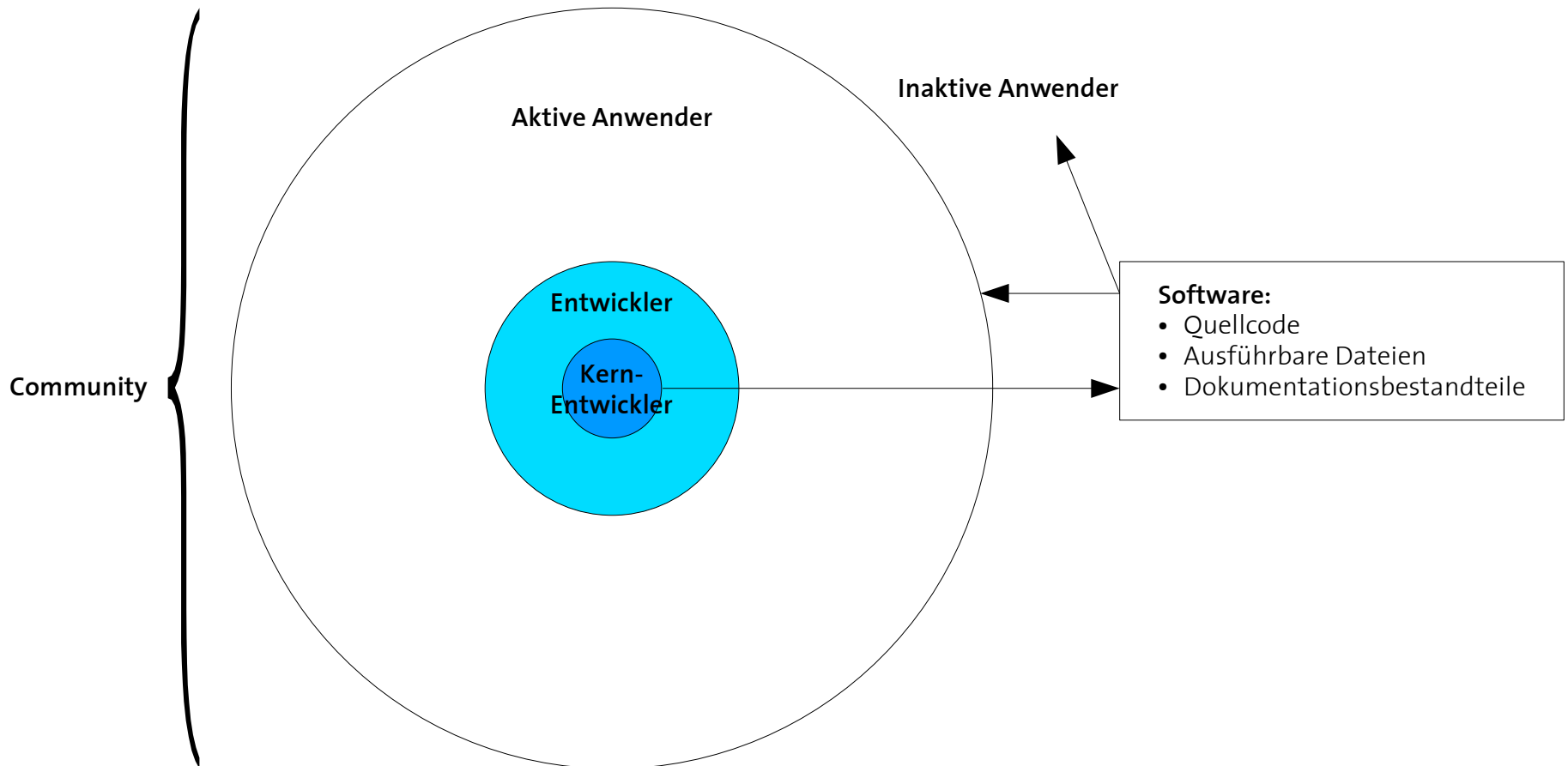
1. **Makro-Ebene:** Best Practices erfolgreicher Projekte
2. **Mikro-Ebene:** Motivation, Anreizsysteme

Untersuchte Open Source Projekte



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

Schalenmodell einer Open Source Community



Charakteristiken idealer 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
- Programmier talent
- 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:** Technische Umgebung verwenden
 - **Physische Begegnungen:** Physische Treffen veranstalten
 - **Trägerorganisation:** Rechtliche Trägerschaft des Projekts gründen
 - **Internationalisierung:** Eine internationale Community anstreben

Empfehlungen pro Gebiet und Ebene

↓ Gebiet	Ebene→	Rekrutierung	Zusammenarbeit	Produktivität
Modularität		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
Dokumentation		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
Release Management		Regelmässig häufig Software-Releases inklusive aussagekräftiger Ankündigungen herausgeben	Fairer Veröffentlichungsprozess, u.a. mit Feature Freeze, praktizieren	Rückwärtskompatibilität gewährleisten und wenn nicht möglich Migrationskripte entwickeln
Kollaborationsplattformen		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
Physische Begegnungen		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
Trägerorganisation		Das Projekt einer bestehenden Träger-schaft anschliessen oder eine neue gründen	Organisatorische, rechtliche und repräsentative Aufgaben durch die Träger-schaft wahrnehmen	Entwicklertätigkeiten durch die Träger-schaft unterstützen und beschützen
Internationalisierung		Auf Englisch kommunizieren und Übersetzungsaufgaben ausschreiben	In einer multikulturellen Community höflich und respektvoll miteinander kommunizieren	Mehrsprachigkeit der Software auf technischer Ebene von Anfang an vorsehen

Schlussfolgerungen Makro-Ebene

- Struktur folgt Problemen → Reaktion, nicht Pro-Aktion
- Offenheit für neue Mitwirkende, andere Ideen und nachfolgende Koordinationspersonen
- Anreize einführen für spezifische Aufgaben, die nicht 'automatisch' erledigt werden

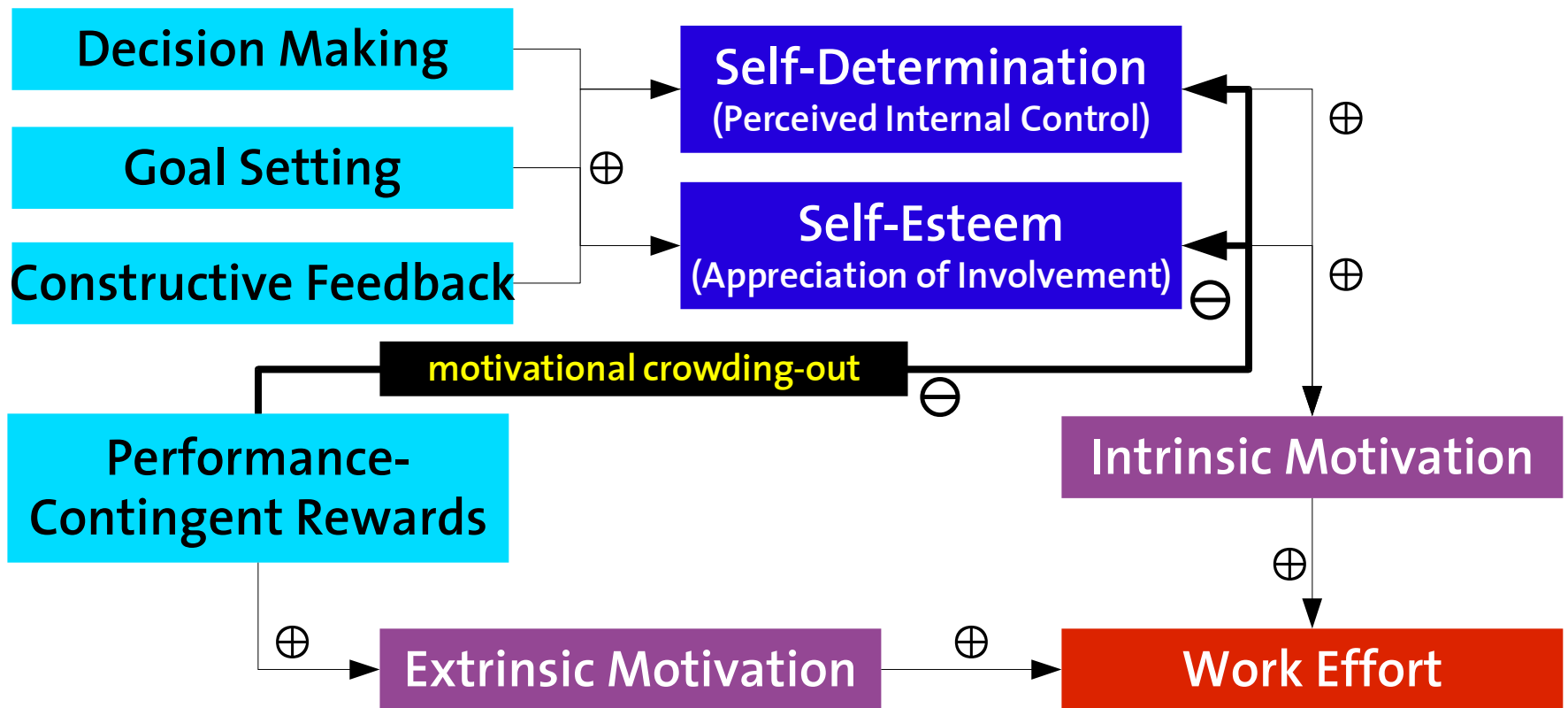
Übersicht

1. **Makro-Ebene:** Best Practices erfolgreicher Projekte
2. **Mikro-Ebene:** Motivation, Anreizsysteme

Intrinsische vs. extrinsische Motivation

- **Intrinsische Motivation** (aus der Aufgabe selber)
 - Basierend auf Freude an der Arbeit
 - Basierend auf innerer Verpflichtung
- **Extrinsische Motivation** (ökonomische Präferenzen)
 - Nicht-monetäre Anreize: Reputation, Karriere...
 - Monetäre Anreize: Gehalt, Kundenaufträge, Boni...

Effects of Motivational Incentives on Effort



Source: Weibel et al. 2007; Frey and Jegen, 2000

Motivationsmix in Open Source Communities

■ Intrinsische Motivation

- Spass, Neugierde, Ideologie („Software muss frei sein.“)
- Verantwortungsgefühl

■ Extrinsische Motivation

- Effizienzvorteil, Lerneffekt
- Lohnvorteil, Bonus
- Vorschrift von Vorgesetzten

Unattraktive Aufgaben in OSS Projekten

- Was wird ausgeführt? → **Was die Entwickler stört.**
 - Was bleibt auf der Strecke? → **Usability**
 - Hochwertige Dokumentation
 - Grafische Oberfläche
 - Funktionen für Endbenutzer
 - Was bleibt auf der Strecke? → **Qualität**
 - Code Review
 - Fehlerkorrekturen
- „**Letzte Meile**“ wird oft vernachlässigt

Extrinsische Anreize in Open Source Projekten

■ Nicht-monetär

- Danksagungen, Anerkennung, Autorenschaft
- Credit Point System (pro Fehlerkorrektur, Doku etc.)
- Aktivitäts-Ranking

■ Monetär

- Anstellung von Kern-Entwicklern
- Bounty System
- Sponsoring von Projekten
- Wettbewerbe

Schlussfolgerungen Mikro-Ebene

- Engagement in Open Source Communities ist ein Mix aus intrinsischen und extrinsischen Anreizen
 - Spezifische Anreize für unattraktive Aufgaben schaffen
- Grundsätzliches Ziel: Anreize für Trittbrettfahrer schaffen, damit sie in der Community mithelfen