Open Source Software Development

Matthias Luder

Inhalt

- Open Source Software
- Vergleich mit agilen Methoden

Definition von Open Source

Introduction

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of The Author's Source Code

The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Restrict Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

***10. License Must Be Technology-Neutral**

No provision of the license may be predicated on any individual technology or style of interface.

Definition von Open Source

- OSD erlaubt Weiterverteilen von Software
- Verkauf nicht ausgeschlossen
- Zugang zum Quellcode gewährleisten
- Code darf verändert werden, welcher wiederum weitergegeben werden darf
- Anwendungszweck nicht eingeschränkt
- Lizenzmodelle aus der OSD abgeleitet, zB. GPL, BSD, usw.

Historischer Abriss

- 80er Jahre:
  - Bill Joy
    - BSD
    - Unix
  - Donald Knuth
    - TeX
  - Richard Stallmann
    - Free Software Foundation
    - GNU

Historischer Abriss

- 80er Jahre:
  - Rechenzeit und Telekommunikation sind teuer und stehen nur begrenzt zur Verfügung
  - Telematiksektor als treibende Kraft für OSS

Historischer Abriss

- 90er Jahre:
  - Linus Torvalds
    - Freax
    - Linux
  - Apache
  - Mozilla

Historischer Abriss

- 90er Jahre:
  - Rechenzeit und Telekommunikation sind erschwinglich (für Studenten)
  - Universitäten beteiligen sich an OSSD
  - Internet als treibende Kraft

Historischer Abriss

- Heute:
  - OSS mainstream
  - Gestande Softwarefirmen unterstützen OSS Development (IBM, Apple, Dell, Corel, Sun Microsystems, usw.)
  - Reine OSS Firmen leben teilweise immer noch (Red Hat, VA Linux, Sleepycat, usw.)

„The Cathedral ...“ (Eric S. Raymond)

„... and the Bazaar“ (Eric S. Raymond)

Projekt Lebenszyklus

(1) „scratching an itch“ und erster Entwurf

(2) Haben andere gleiche Probleme?

(3) Wissensaustausch mit anderen, es entsteht ein erstes vages Bild

(4) Start eines informellen Projekts

Projekt Lebenszyklus

(5) Arbeit an einem Teilproblem

(6) Produkte öffentlich zugänglich machen

(7) Verbesserungsvorschläge fliessen ein

(8) Es machen neue Entwickler mit

(9) Noch mehr Entwickler und Wissen

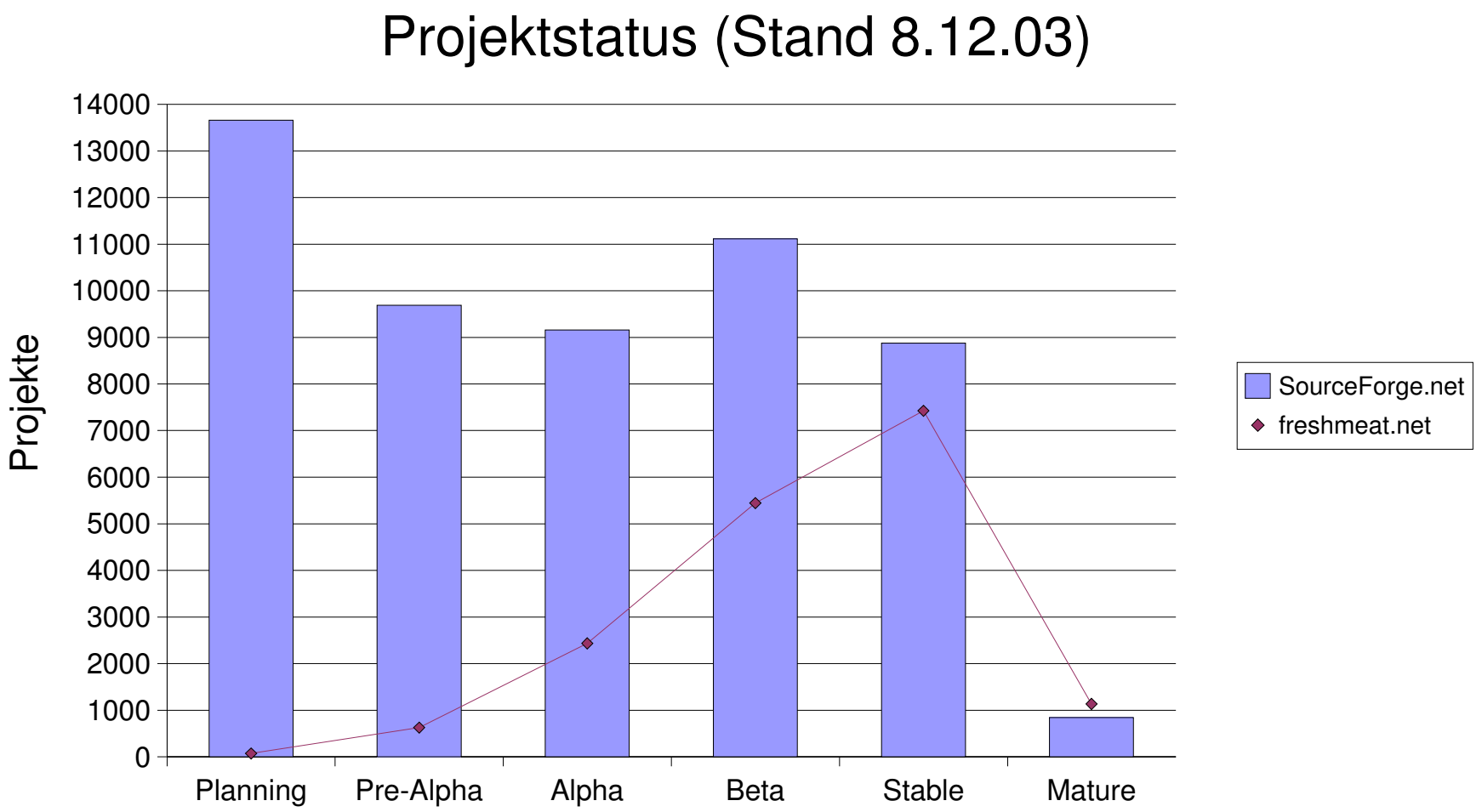
(10) Ein Zyklus ist geschlossen -> Punkt (5)

(11) Projektgemeinschaft entsteht

Klassifikation

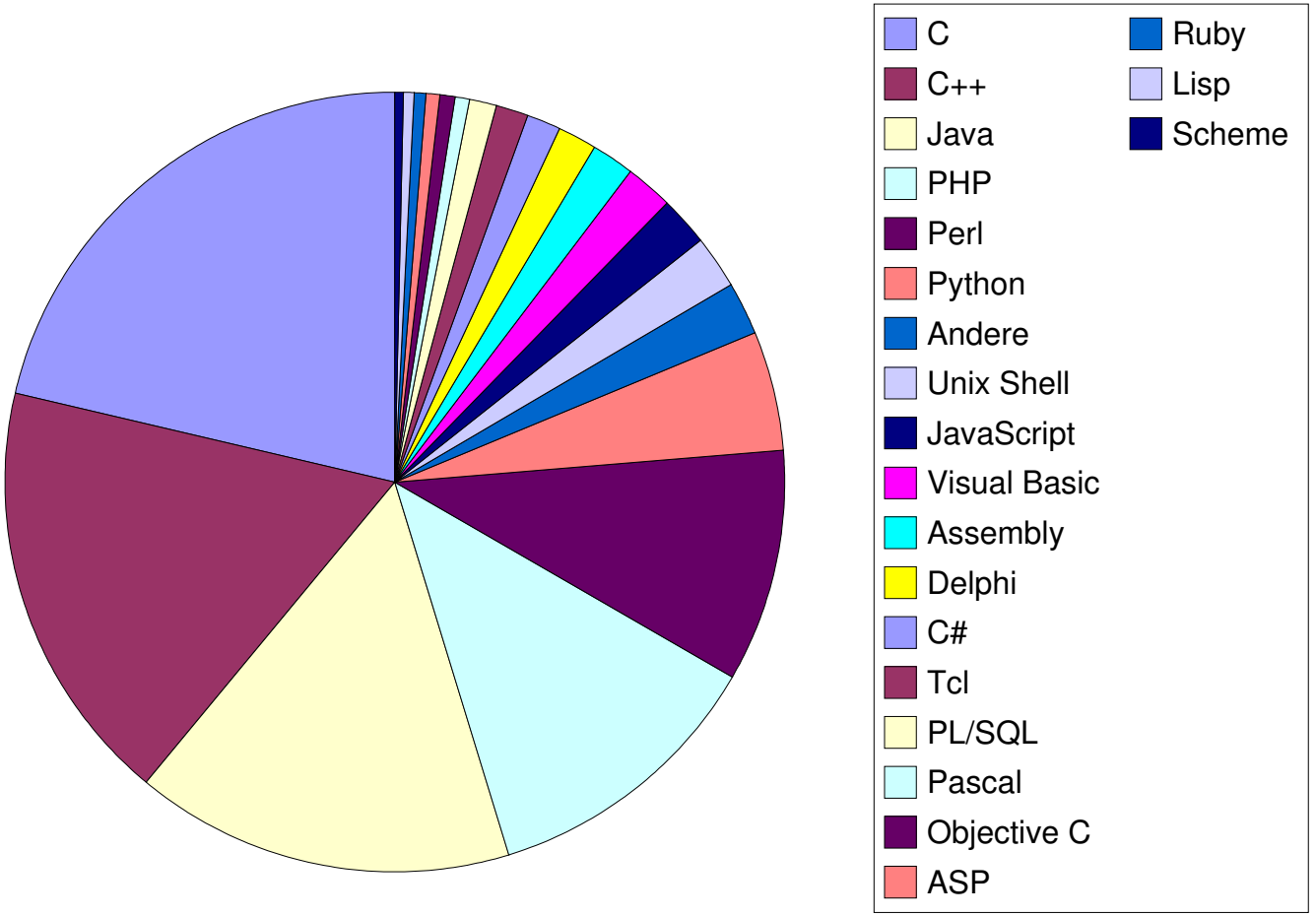
- Planning
- Pre-Alpha
- Alpha
- Beta
- Stable
- Mature

Beispiel SourceForge.net und freshmeat.net



Beispiel SourceForge.net und freshmeat.net

Programmiersprachen (Stand 8.12.03)



Prozesse und Organisation

- Die Anwendung von klassischen Methoden kann nicht ausgeschlossen werden
- Der Entwickler wählt (im Idealfall) die Problemstellung, nicht der Kunde
- Entwicklung in kleinen Gruppen, meistens mit einem (Projekt-)Leiter
- Inkrementelles Wachstum

Prozesse und Organisation

- Anforderungen von Kunden fliessen erst relativ spät in die Entwicklung ein
- Geographische Distribution bei grossen Projekten
- Formlose Entwicklung; keine Prozessdokumentation
- Kein Gesamtprojektdesign
- Gemeinsames Interesse muss zu jeder Zeit vorhanden sein

Agile Methoden im Vergleich mit Open Source

- Festlegen der Methoden in Open Source Entwicklung fast nicht möglich
- Viele verschieden Prozesse hinter OS
- Prozesse und Methoden sind bei OS nicht dokumentiert
- Allgemeiner Vergleich nur auf konzeptioneller Ebene möglich

Prinzipien der agilen Methoden

- Je kleiner die Teilaufgaben und je kürzer die Rückkopplungszyklen desto besser
- Der Kunde gestaltet das Produkt während seiner Entstehung
- Je freier und konzentrierter die Entwickler arbeiten können, desto besser
- Die Qualität wird an der Quelle sichergestellt
- Keine Arbeit auf Vorrat

Kleine Teilaufgaben und kurze Rückkopplungszyklen

- „Release early, release often“ (z.t. täglich)
- Durch dezentrale Entwicklung zu inkrementellem Wachstum gezwungen
- Paralleles Entwickeln möglich
- Hohe Kohäsion der einzelnen Module

Der Kunde gestaltet das Produkt

- Mitarbeit muss nicht explizit verlangt werden, es geschieht freiwillig
- Kunde = Entwickler
- Zusammenarbeit vs. Machtkämpfe

Freies und konzentriertes Arbeiten der Entwickler

- Grösste Freiheit auf dem „Bazaar“
- Motivation
  - Persönliche Herausforderung
  - Anderen helfen
- Intresse kann sich allerdings sehr schnell wieder abnehmen

Qualität an der Quelle sicherstellen

- Benutzer stellt Qualität sicher (Co-Entwickler, Debugging)
- Qualität von OSS wird als hoch angesehen
  - Relativ späte Stable Releases
  - Kein Zeitdruck
  - Der Kunde hat (geringen) Einfluss
  - Erfahrene Entwickler ...?

Keine Arbeit auf Vorrat

- Restrukturierung bei Bedarf
- Paralleles Entwickeln
- Heterogenität zwingt zu allgemeingültigen Lösungen

Endbetrachtung

- Parallelen auf allgemeinem Niveau
- Der Entwickler als Mensch im Zentrum
- Detaillierte und langfristige Planung fehlen, massgeben für die Form der Prozesse
- Kunde und Entwickler unterscheiden sich bei OSS fliessend

Endbetrachtung

- Agile Methoden „best practice“ Ansätze in einer klassischen Umgebung
- Unterschiedliche Rahmenbedingungen

Endbetrachtung

- Open Source Software Entwicklung kann nicht als agile Methode angesehen werden