the gnome bazaar

how gnome gets built and how we can improve

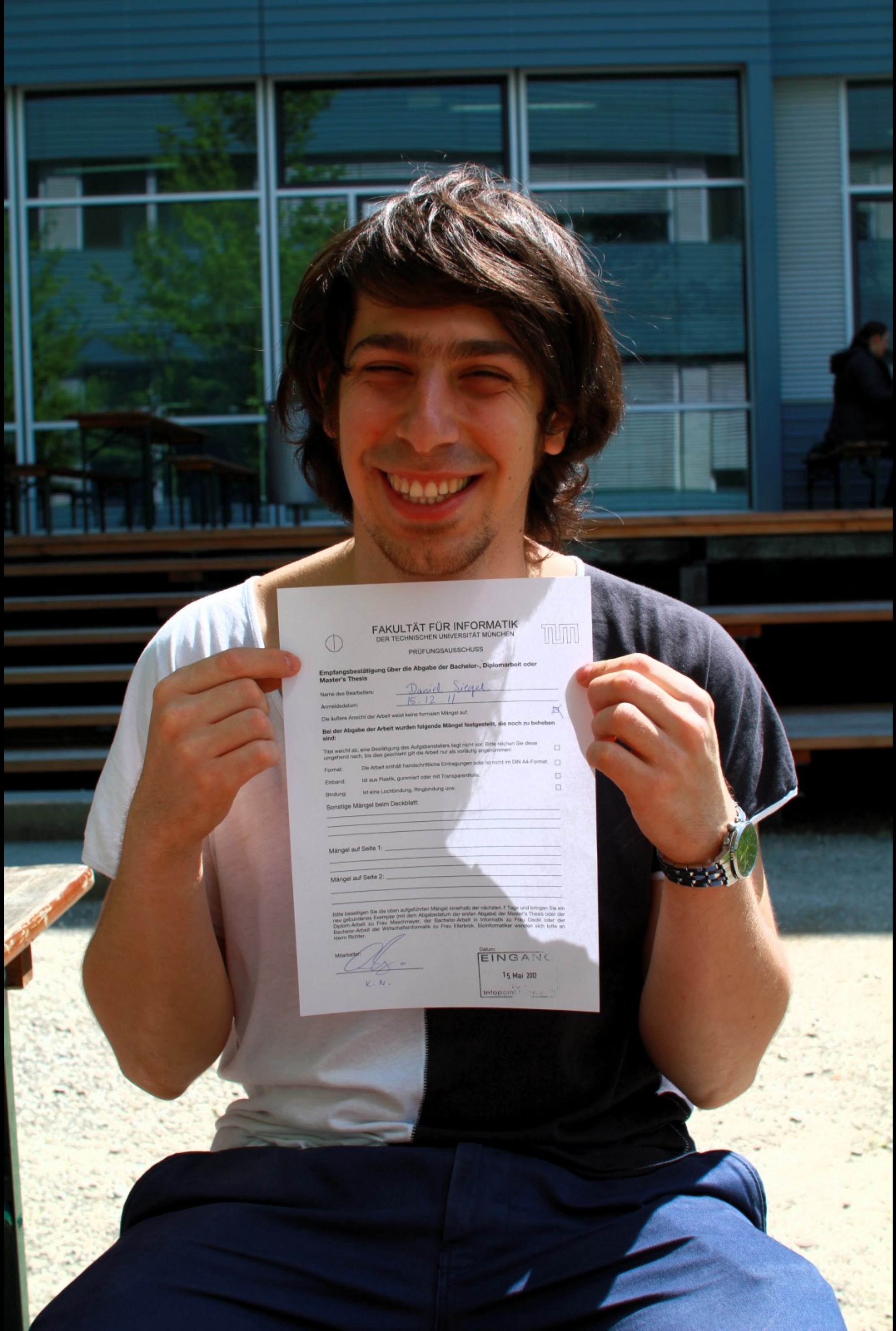
daniel g. siegel

Fakultät für Informatik
Technische Universität München

Master's Thesis in Informatics

Typical Development Processes of Free and Open Source Software Projects

Daniel Siegel



FAKULTÄT FÜR INFORMATIK
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

PRÜFUNGSAUSSCHUSS

Empfangsbestätigung über die Abgabe der Bachelor-, Diplomarbeit oder Master's Thesis

Name des Bearbeiters: Daniel Siegel

Anmeldedatum: 15.12.11

Die äußere Ansicht der Arbeit weist keine formalen Mängel auf. ☒

Bei der Abgabe der Arbeit wurden folgende Mängel festgestellt, die noch zu beheben sind:

Titel weicht ab, eine Bestätigung des Aufgabenstellers liegt nicht vor. Bitte reichen Sie diese umgehend nach, bis dies geschieht gilt die Arbeit nur als vorläufig angenommen! ☐

Format: Die Arbeit enthält handschriftliche Eintragungen oder ist nicht im DIN A4-Format. ☐

Einband: Ist aus Plastik, gummiert oder mit Transparentfolie. ☐

Bindung: Ist eine Lochbindung, Ringbindung usw. ☐

Sonstige Mängel beim Deckblatt:

Mängel auf Seite 1:

Mängel auf Seite 2:

Bitte beseitigen Sie die oben aufgeführten Mängel innerhalb der nächsten 7 Tage und bringen Sie ein neu gebundenes Exemplar (mit dem Abgabedatum der ersten Abgabe) der Master's Thesis oder der Diplom-Arbeit zu Frau Maschmeyer, der Bachelor-Arbeit in Informatik zu Frau Oeckl oder der Bachelor-Arbeit der Wirtschaftsinformatik zu Frau Ellerbrok. Bioinformatiker wenden sich bitte an Herrn Richter.

Mitarbeiter: [illegible]
K. N.

Datum:
EINGANG
15. Mai 2012
Infopoint [illegible]

1. some serious stuff about my thesis
2. awesome gnome stuff

how do foss projects work, which structures do they have and which workflows have they established. to accomplish this, several foss will be analyzed in order to identify concertedly models. in addition they will be compared to traditional software engineering models in order to see whether they are similar or oppose differences.

good selection of projects with which the analysis is able to produce reliable and reasonable results

- popularity
- age
- category
- activity
  - releases
  - downloads
  - commits
- community
  - communication
  - number of developers
  - conferences
  - foundations
  - ongoing projects

project	origin	category
Debian	1993	operating system
Drupal	2001	content management system
Fedora	2002	operating system
GNOME	1997	desktop environment
KDE	1996	desktop environment
MySQL/MariaDB	1997	database management system
PHP	1994	interpreted programming language
Plone	1999	content management system
PostgreSQL	1986	database management system
Python	1989	interpreted programming language

results

1 • history & origin

2 • community structure

3 • release process

4 • development model

"[...] rather, the community seemed to resemble a great babbling bazaar of differing agendas and approaches"

eric s. raymond

what?

comparison

1 • history & origin

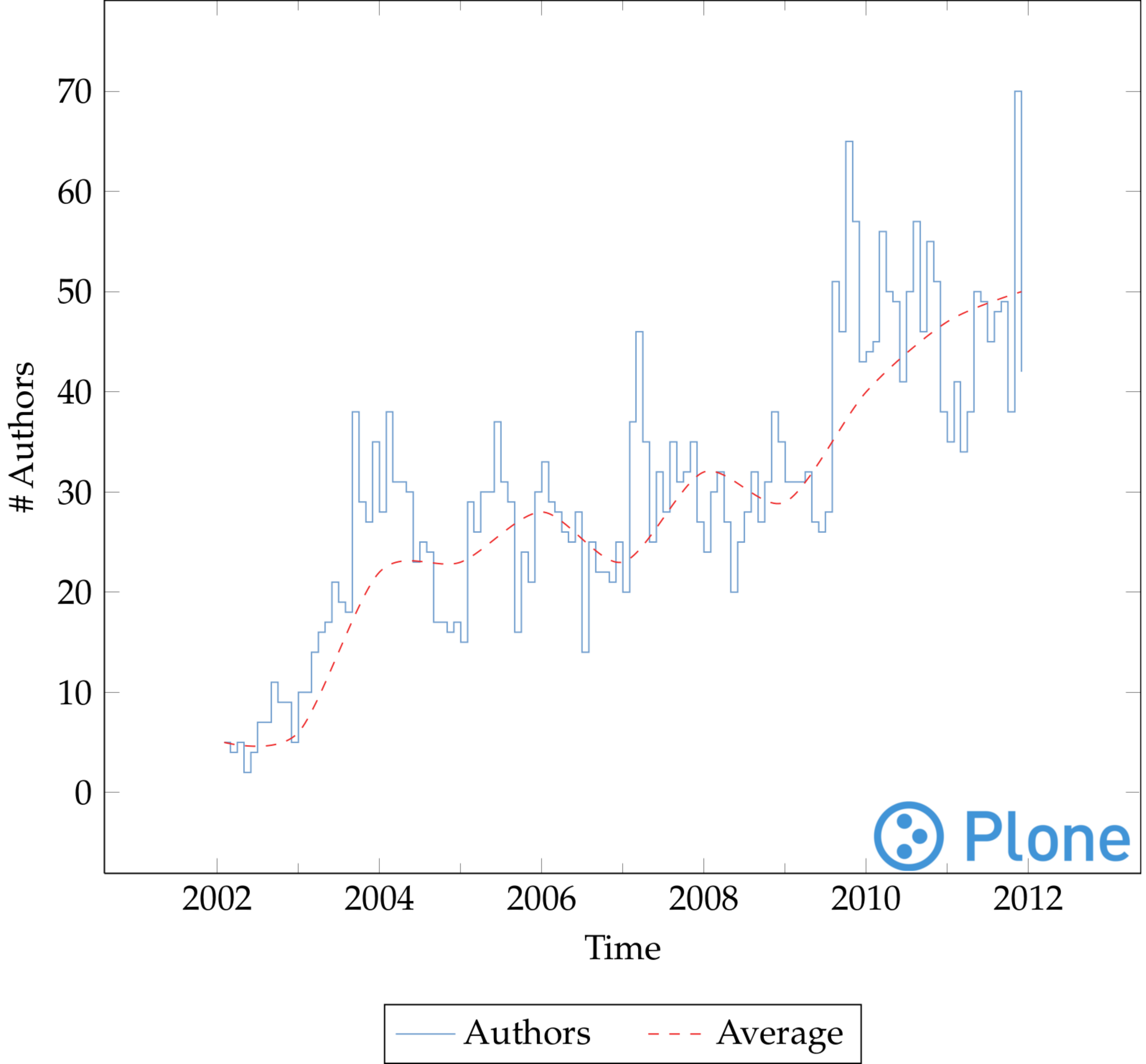
2 • community structure

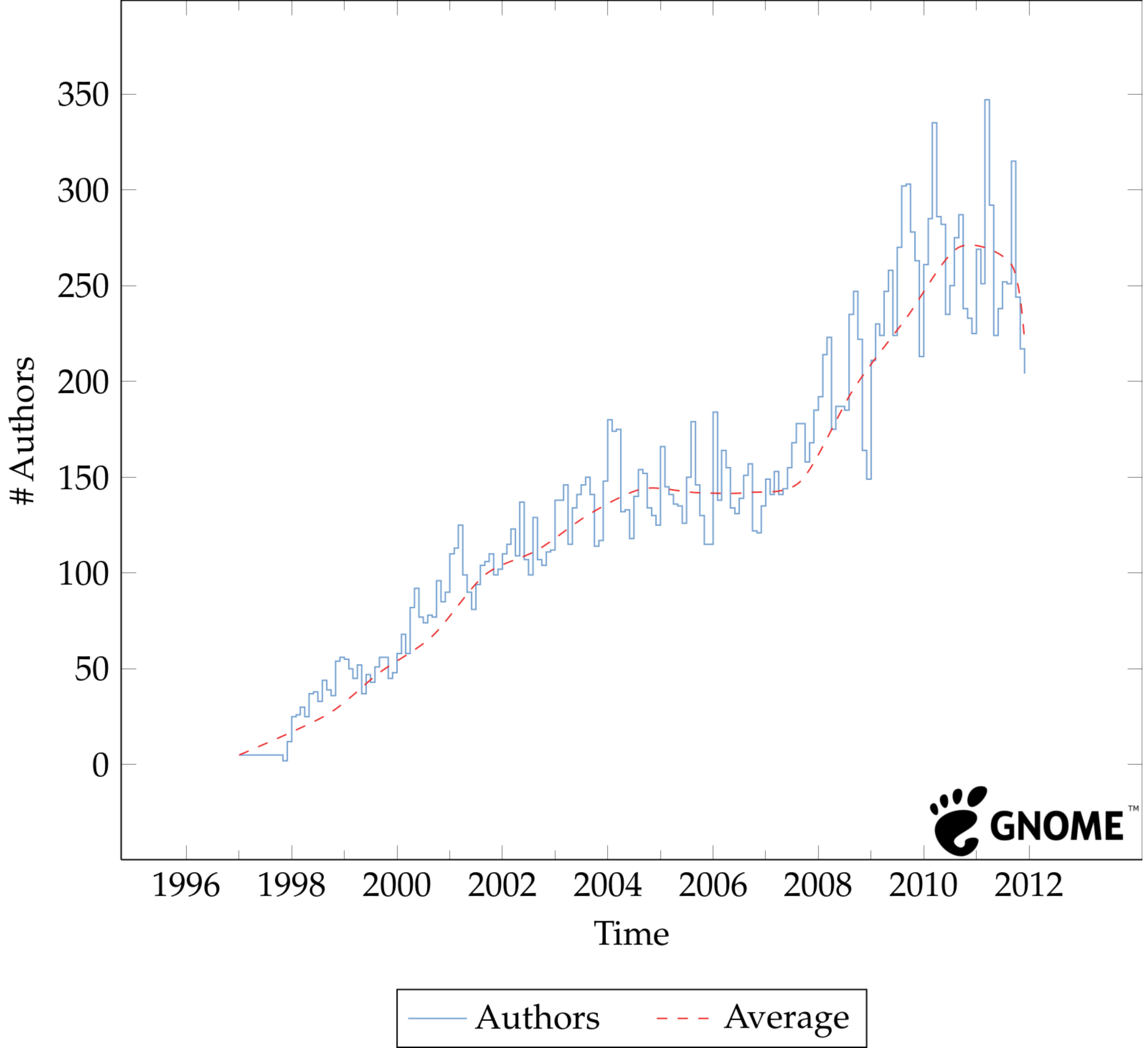
3 • release process

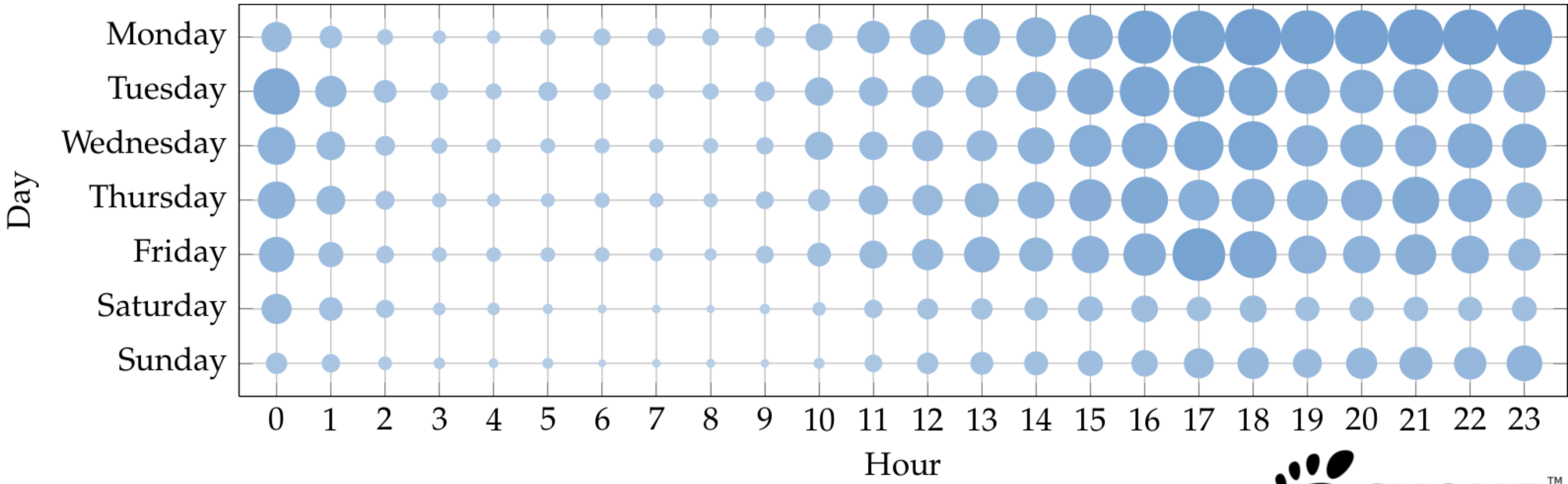
4 • development model

history & origin

- diverse origin
- small group of founders
- big burst of growth after first release
- more big bursts before big releases


70
60
50
40
30
20
10
0
# Authors
2002
2004
2006
2008
2010
2012
Time
Authors
Average
Plone

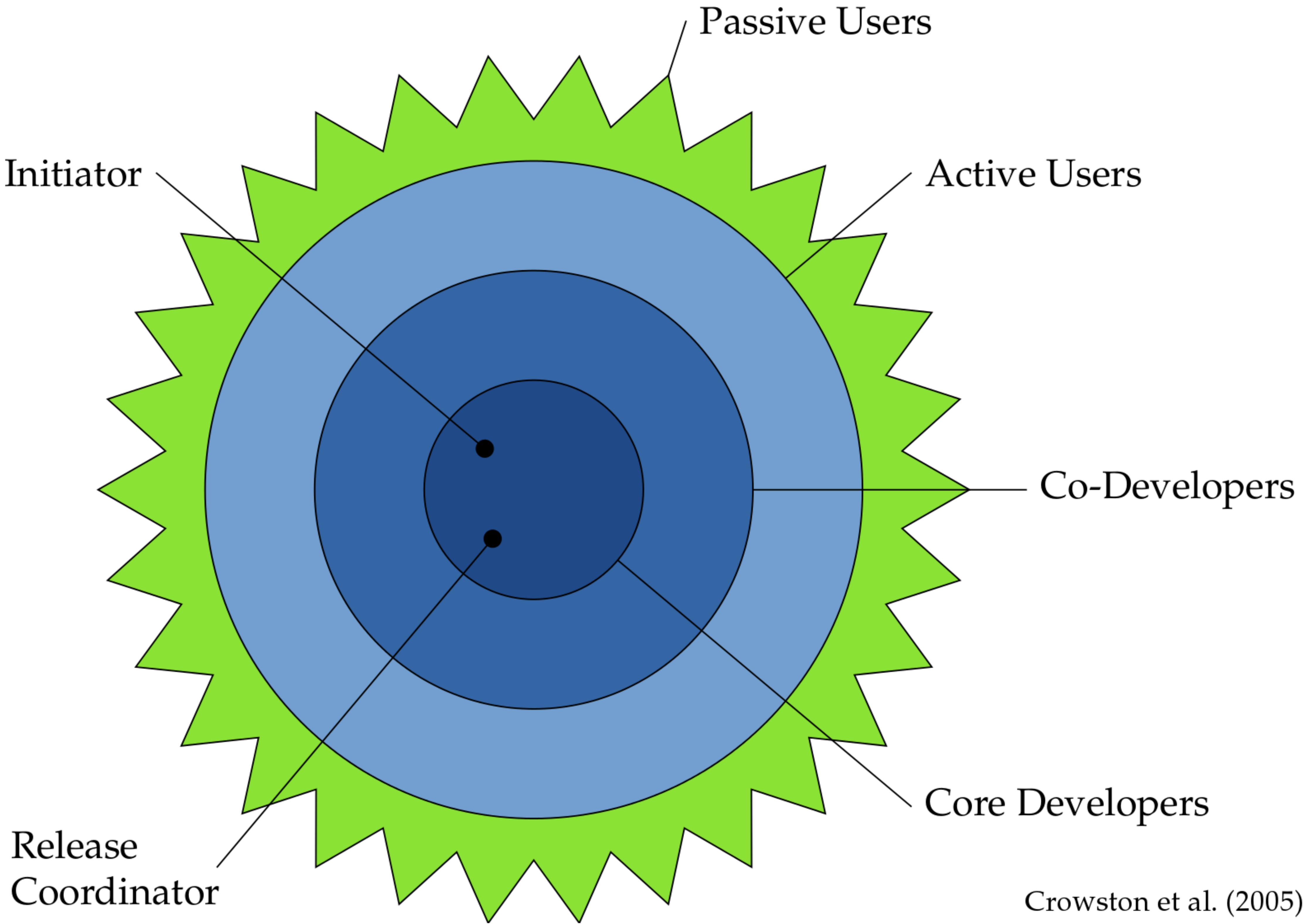

350
300
250
200
150
100
50
0
# Authors
1996
1998
2000
2002
2004
2006
2008
2010
2012
Time
GNOME™
Authors
Average

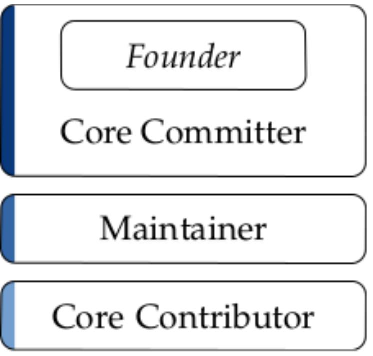

Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Day
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
Hour

GNOME™

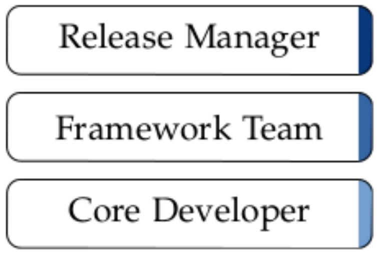
community structure

- very hierarchical
- lead by leader or team
- differences in hierachical structure
- though easy to step up the ladder

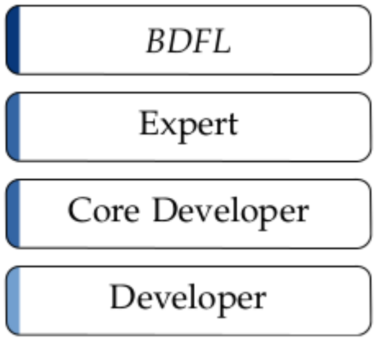

Passive Users
Initiator
Active Users
Co-Developers
Core Developers
Release Coordinator
Crowston et al. (2005)



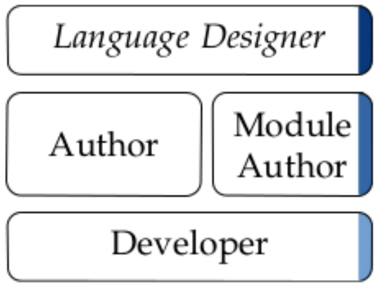
(a) Drupal



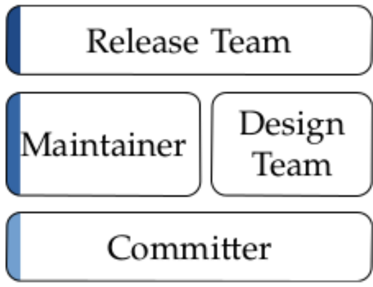
(b) Plone



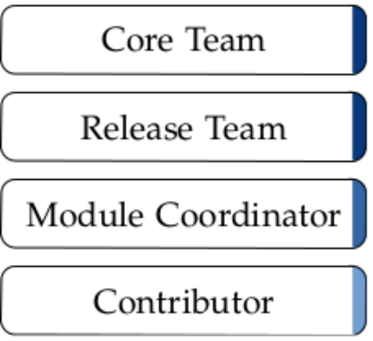
(c) Python



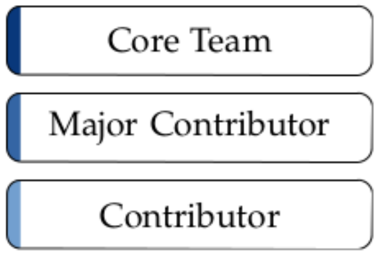
(d) PHP



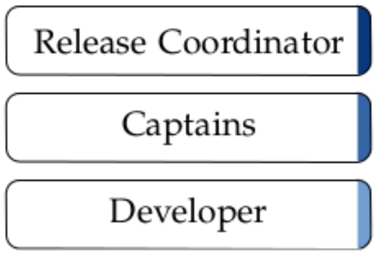
(e) GNOME



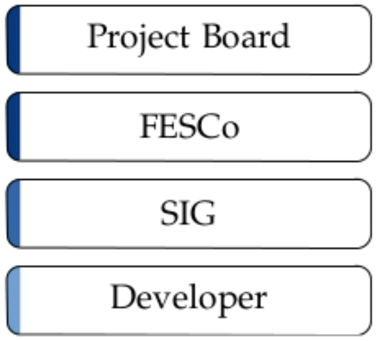
(f) KDE



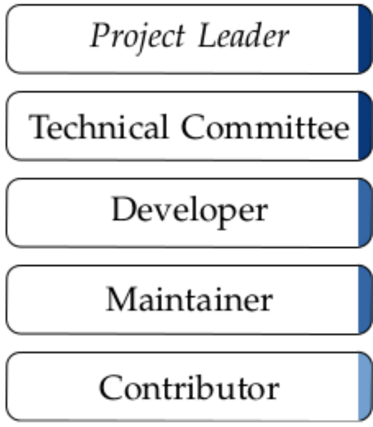
(g) PostgreSQL



(h) MariaDB



(i) Fedora



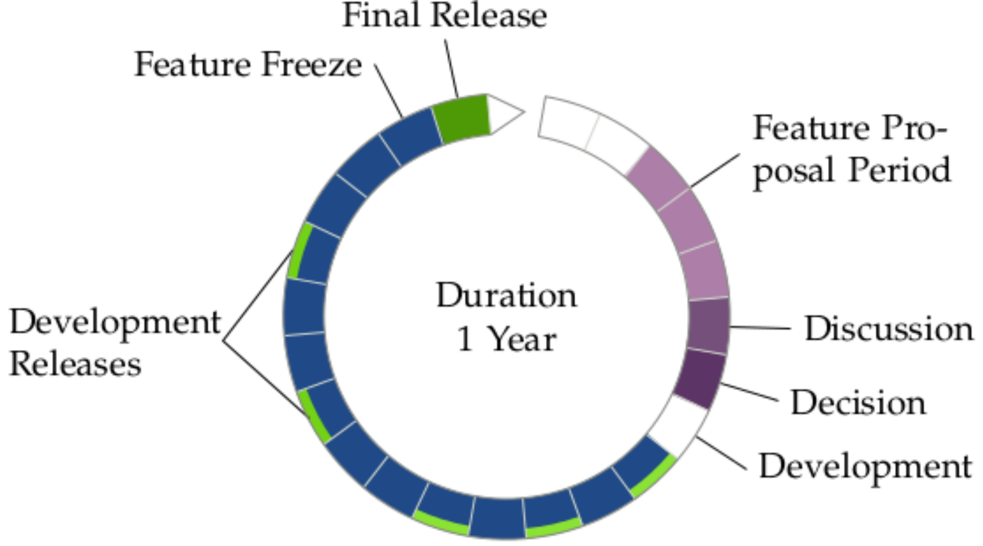
(j) Debian

community structure: remarks

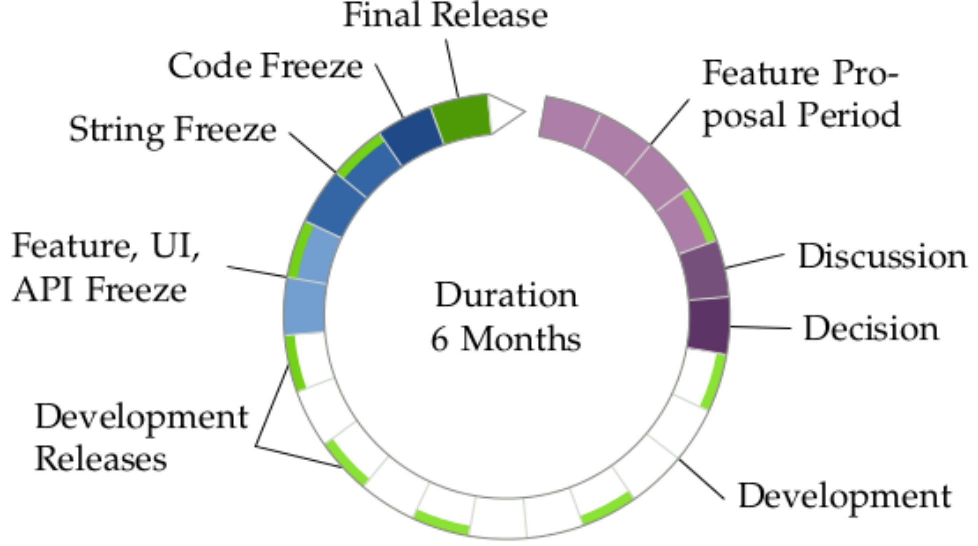
- missing visionary
- role of rt
- unfruitful discussions

release process

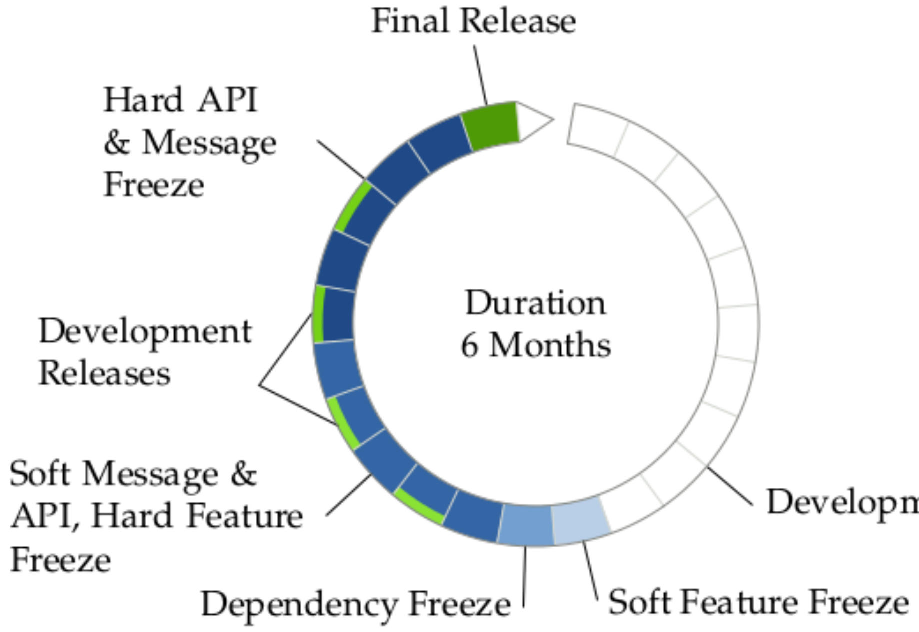
- mostly fixed release cycles
- lead by release manager/team
- similar phases in all projects



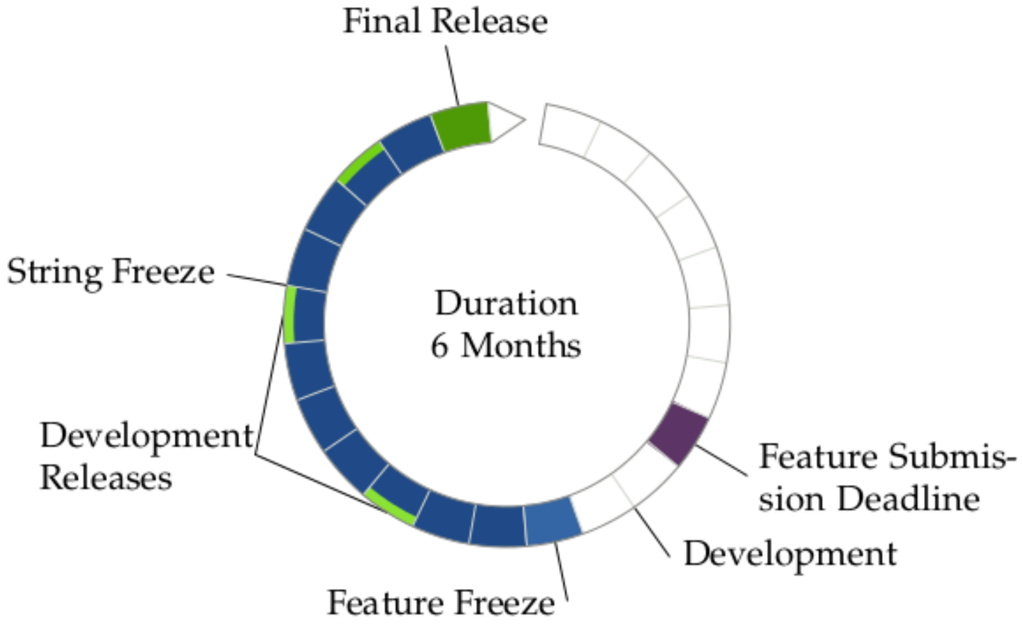
(a) PHP



(b) GNOME



(c) KDE



(d) Fedora

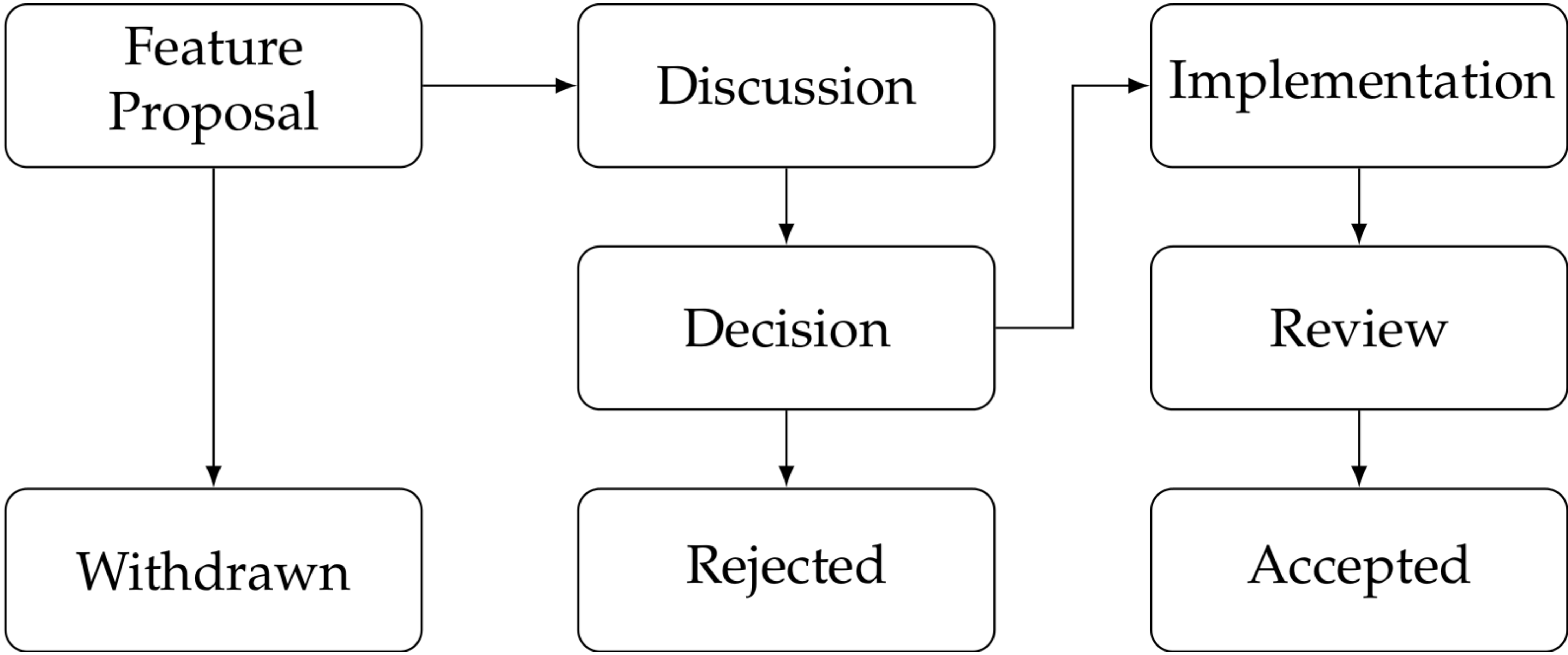
release process: remarks

- cycle often too long for small projects
- api/abi compatibility
- jhbuild etc. needed

[missing: some boring slides about software engineering and development models]

features development

- similar feature inclusion processes
- range from dynamic to very structured
- established in all projects


Feature Proposal
Discussion
Implementation
Decision
Review
Withdrawn
Rejected
Accepted

features development: remarks

- a bit more structure would be great
- somewhat intransparent decision making
- short and adverse placed period

that's all folks!