UMONS
Université de Mons

An empirical study on the specialisation effect in Open Source communities

Mathieu Goeminne

Faculté
des Sciences

Agenda

- Previous research
- Current research

Previous research

What?

- Collaboration between Alexander Serebrenik, Bogdan Vasilescu, Tom Mens and Mathieu Goeminne
- Study of the Gnome ecosystem
- Analysis of its **activity** and the **specialisation** of its members.

Main goal: understand how the workload varies across projects, persons, and activity types in the ecosystem.

Activity type

Type of change made on the source code

Ex: coding, localisation, documentation, ...

(cf. *Beyond source code: The importance of other artifacts in software development*, Robles *et al.*)

Main difference: an activity type is associated to each touched ***file*** *(not to each* ***commit****).*

Activity definition

- Set of rules, each rule has an activity type and a regular expressions. Example: (‘coding’, ‘.*\.c’)
- If the full path of a file matches the regex, the corresponding activity type is associated to the file.
- A file can match many rules → rules are ordered.

Workload and Involvement Metrics

- APT**W**(p,d,t) = # files in *p* touched by *d* and having *t* for activity type.
- APT**I**(p,d,t) = 1 if APTW(p,d,t) > 0

  0 otherwise

Workload and Involvement Metrics (cont.)

- Higher level metrics : aggregation over all projects, developers and/or activity types. Ex:
  - NPD(d) = Number of projects in which d is involved.
  - PW(p) = Global project workload.
  - RPTW(p,t) = Workload in *p* for *t*, relative to the project workload.
  - PWS(p) = imbalance of workload across project activity types (using Gini index).

Some results

- *Coding*, *localisation*, *development documentation* and *system building* are the most important activities.
- Software projects with more activities generate more workload.
- Most projects specialise their workload in a few activities.
- Most developers concentrate their workload in few activities. This specialisation relates to the number of projects in which they are involved.
- If *coding* generates the highest workload share, it does not attract the highest fraction of project contributors: coding is elitist.

UMONS
Université de Mons

Current research

The same with time effects

Faculté
des Sciences

Goals

- To get a better insight by taking into account the time dimension : How do our metrics and their relations evolve over time?
- To observe evolution patterns in the communities activity.
- Case study based on KDE.

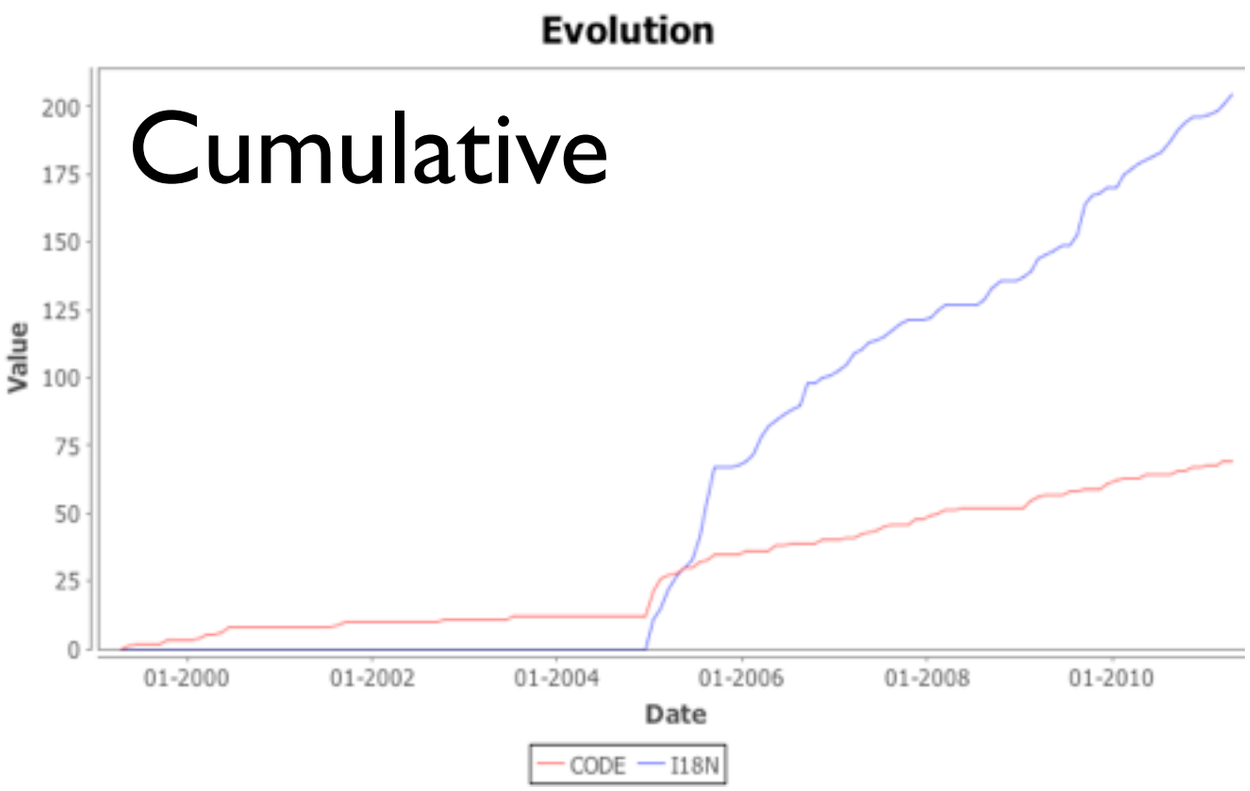
Metrics

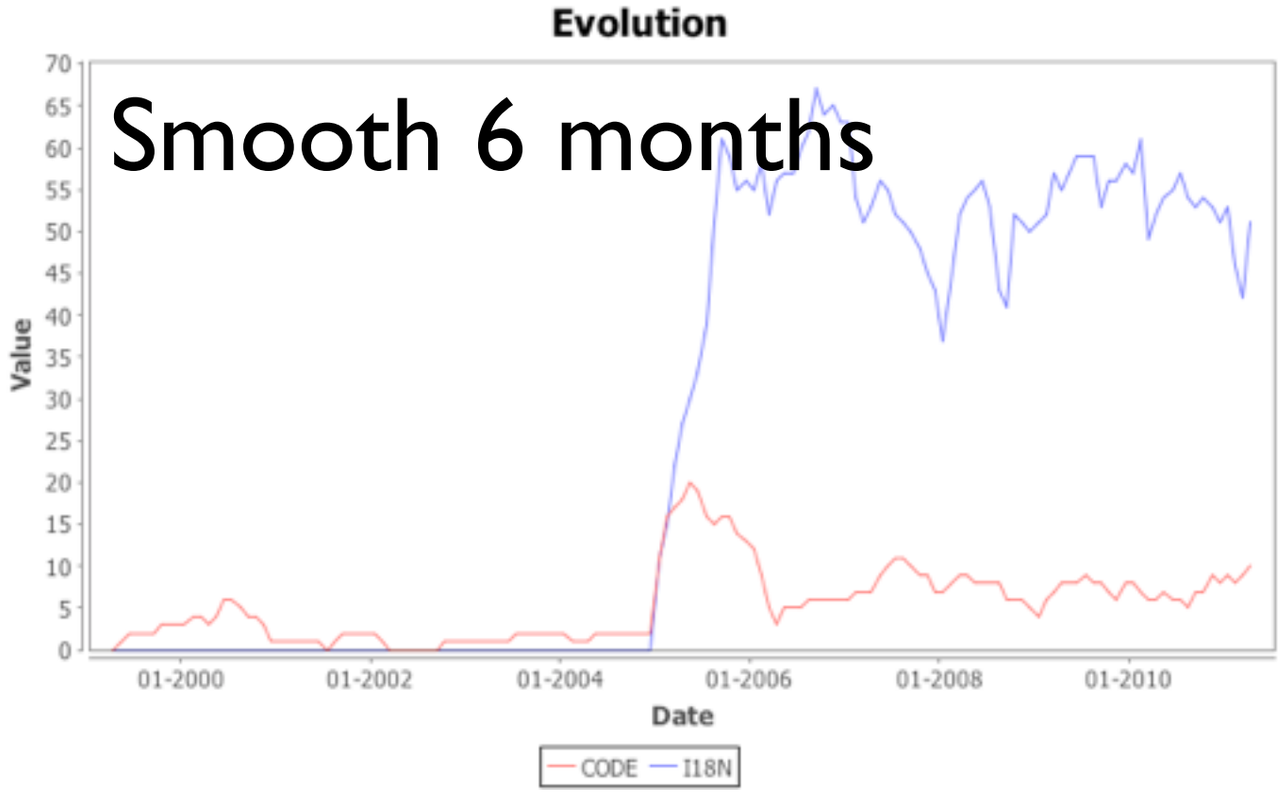
- The same as the ones in our previous work, with a new parameter for the time : APTW(p,d,t,r)
- Filtering/aggregation by project, developer, activity, action (add, remove, change), and time range.
- New, time-based metrics.

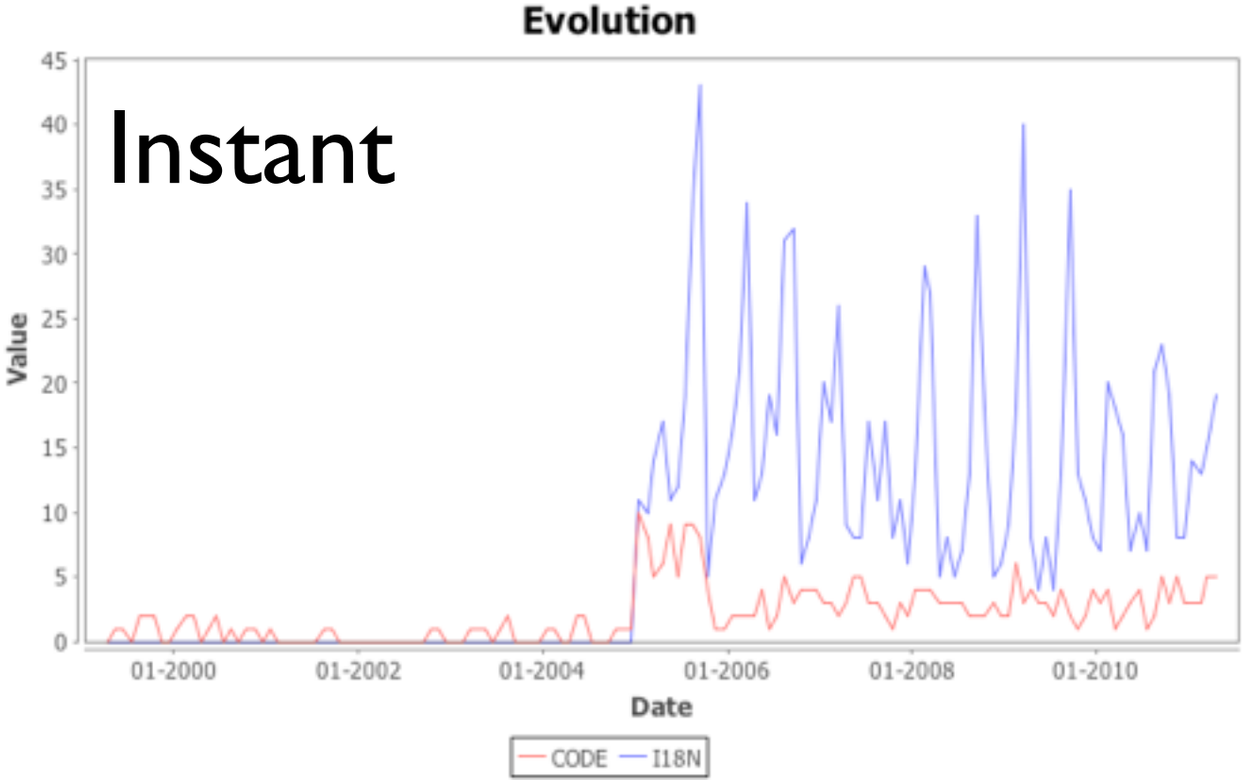
Time range

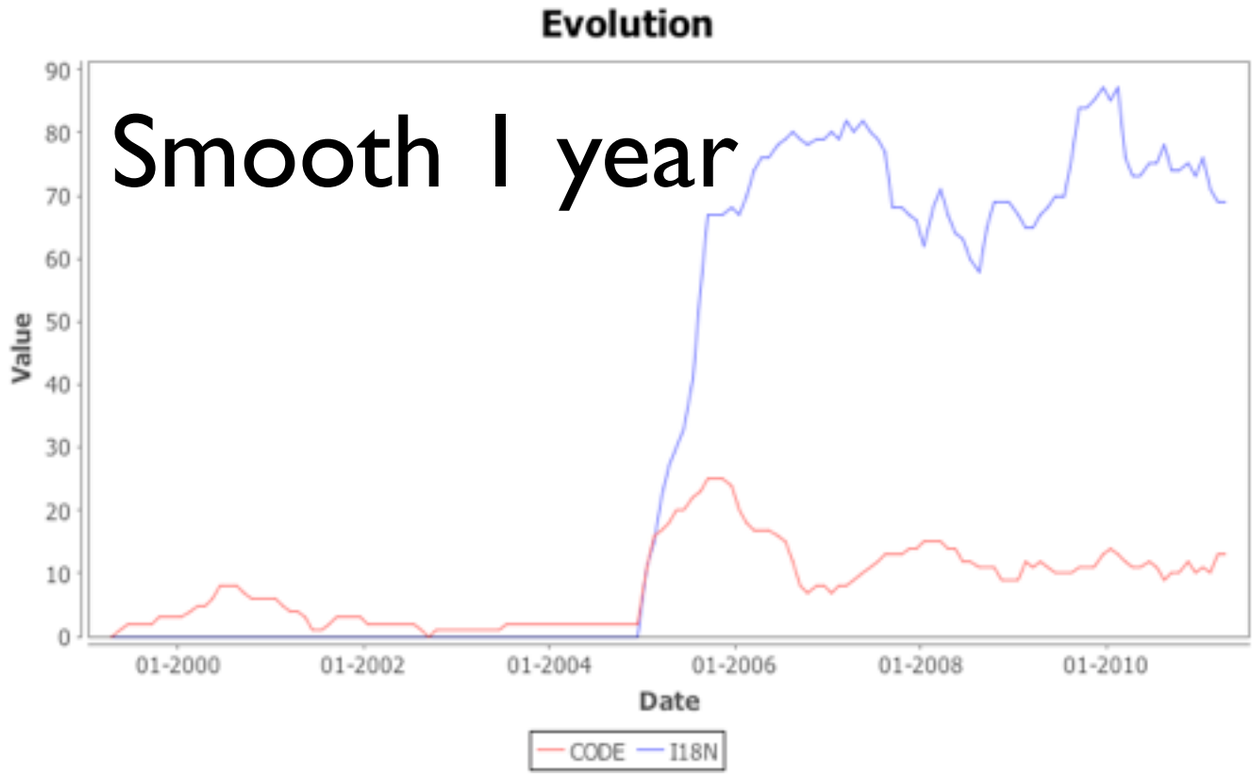
- Characterized by the instant of the measurement and a time range.
- Three ‘modes’:
  - instant: each value covers a different time range.
  - cumulative: each value takes into account the entire history.
  - smoothing: each value partially covers an older one, so that the data are smoothed.

How many active coders and translators in Evince?

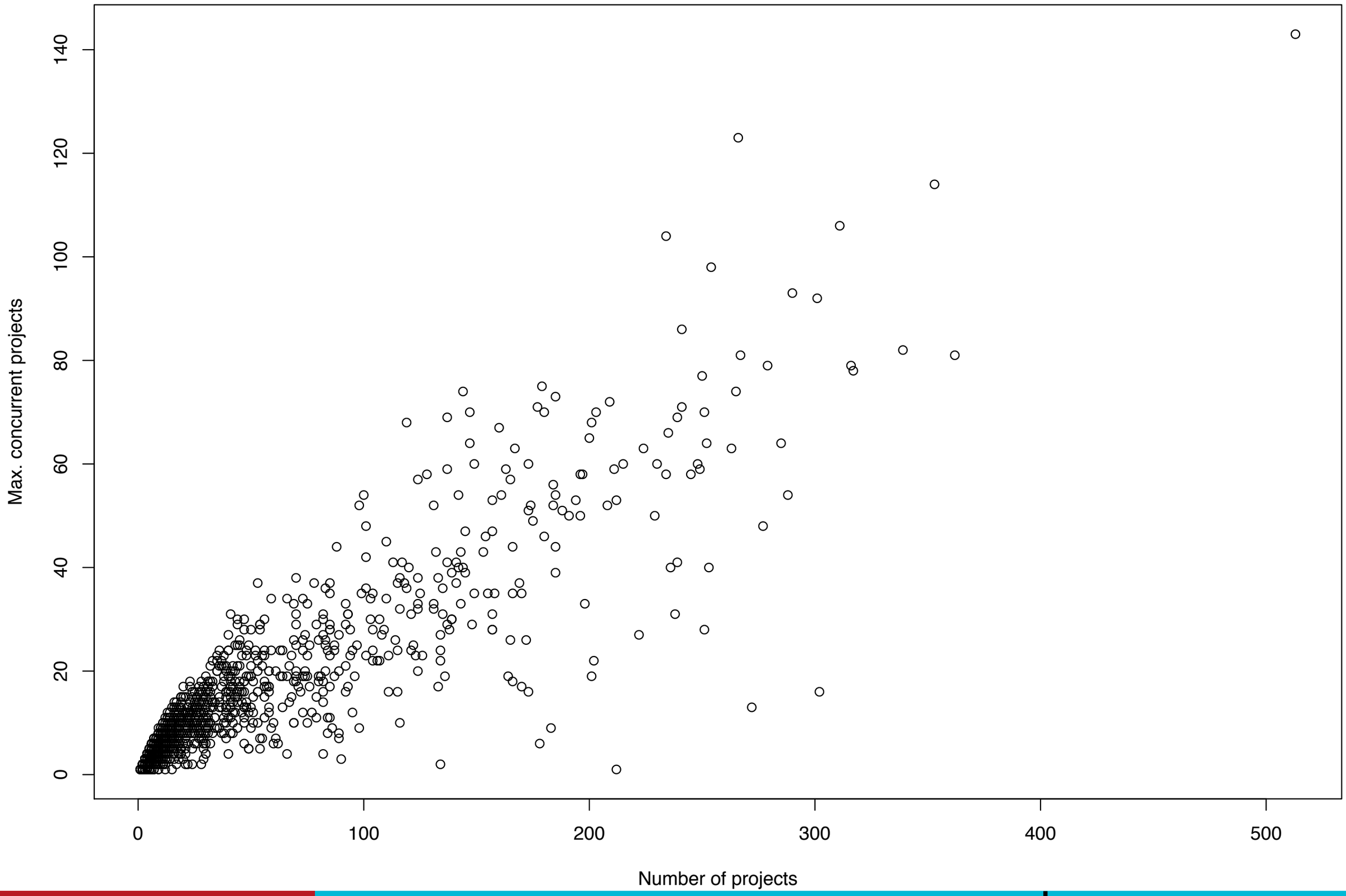




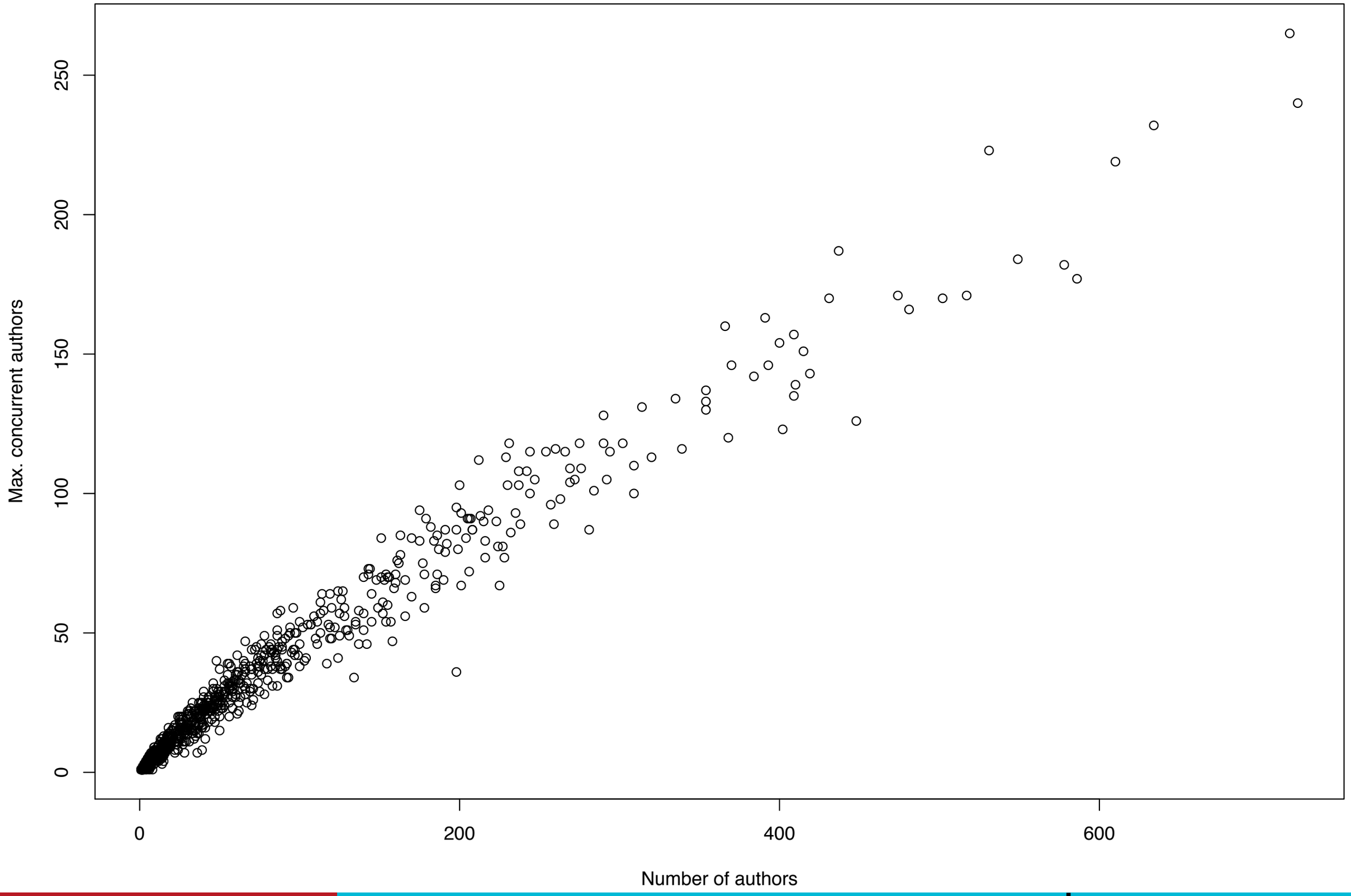




What is the parallel work capacity of Gnome developers?



What is the co-working projects capacity in Gnome?



KDE

- Same behaviors as in Gnome? Not sure for the translation case.
- Based on SVN: all projects are placed in the same repository. Does it support a developers **migration** from a project to another one?
- Can we draw a generic developer's profile

Thank you