



**Johann Wolfgang v. Goethe University**  
**Frankfurt am Main**

**PELM**

**Project Electronic Labour Markets**

# Structure and Organization of KDE

Andreas Brand, Dipl. Soz.  
Institute for Polytechnic und Labor studies  
Department of Social Sciences



## Outline

1. Research question and method of the case study
2. What is KDE? – the structure of the large-scale project
3. The Developers: the inner circle
  - 3a. Definition
  - 3b. Tasks
4. The other groups
5. Summary



## 1. **Research question:**

Does the software production at KDE only work through cooperation?

### **Method of the case study:**

- explorative
- half-structured expert interviews
- analysis of documents, esp. homepages
- quantitative questionnaire (55 answers)
- secondary analysis of 62 interviews (self-information) from KDE-members



## 2. What is KDE? – the structure of the large-scale project

- KDE = large-scale open source project
- approx. 1200-1300 participants; number still growing
- embedded in social movement – hobbyists: focus on technical excellence
- institutional basis: open source code; licence (GPL)



- founded in 1996 at the University of Tübingen
- focal point: production of joint software (desktop und applications)
- several tasks:  
software developers 2/3,  
translators/documentators 1/3,  
graphic/sound <5%, only some exclusive  
homepage-journalists, bugreporters



- important media of communication: mailing lists  
(chat >face-to-face>phone)
- important media of production: versioning  
system on the Internet, bugreport system,  
developer tools



- the large-scale project consists of subprojects for special software themes
- software/projects are central (core libs.)
- in projects: one project leader and several participants
- projects: average 4 persons (min=1; av. max=8; max=20/50)
- work done: average in 2 projects, majority in 1 project (min=0; max=9)



## A typical KDE-participant is:

- young (between 20 and 30 years old; min 16, max 50)
- male
- highly qualified (*went to college/university*)
- a student or employed
- an IT-professional (has an IT- or IT-relevant qualification)
- principally from Germany (worldwide: Europe and USA/Canada)
- partially regional clusters
- single (not married/no kids)
- working voluntarily in his spare time (*only few are paid for work on KDE*)





- Working time: arithmetical mean 15,3/ median 10 hours per week (min 0,5; max 90)
  
- ➔ Different motivations:
  - personal requirements: self-motivation/ preference for developing (bricolage mentality)/intellectual interest
  - supported through teamwork, relations in the community = cooperation



### 3a. The developers: the inner circle –definition





## The inner circle:

- is a loosely coupled group of long-time project participants without leading figures; rest = outer circle
- have a high reputation:  
permanent achievements for the project,  
quality, complexity of the work task,  
expectation of rapid revision of work



- work on important software (core libs.)/mailing lists (visibility)
- important positions as founding-member or takeover through reputation (project leader, release coordinator, project maintaining functions)
  - ⇒ competition for positions through reputation
- cooperative communication expected



## 3b. The Developers: the inner circle – the tasks

### Tasks in KDE:

- Entrance prerequisites = number of code lines, control of quality and test of persistence and  
⇒ self-motivation and control through inner circle
- no influence on work/task distribution, because of voluntary selection according to interest
- production: influence on planning discussion through high experience; help with problems/tasks



- work/task combination: through release coordinator with support of participants (both from inner circle)
- Control:
  - a) self- and peer-review-control; inner circle through random sampling
  - b) compliance with communication/production norms
- **Summary: aim is unity of the project**
  - ⇒ hierarchical element



## Tasks outside KDE:

1. monitoring of norms with respect to licence (GPL) at companies:
  - logos/copyright
  - prompt return of source code
2. contact to other OS-projects with overlapping aims/themes (debian, gnome, ...)



## 4. The other groups

- **Not so important as developers because of distance to source code development**
  - a) Documentators: One person as project leader on top, several subprojects around important software with project leader and participants
  - b) Translators: each language has a translation project with project leader on top level. Projects are divided into subprojects around important software.





## 4. The other groups

- **Both groups depend on self-motivation with cooperative communication.**
- **but are more hierarchical than the developers because only (sub-)project leaders have access to versioning system.**
- **competition for project leader position**



## 5. Summary: How KDE works

- Cooperation (supports self-motivation)
- Hierarchy (coordination of the whole project; no instructions)
- Competition (motivation for reputation, occupation of a small amount of positions)