Overview on trust in large FLOSS communities

Etiel Petrinja, Alberto Sillitti, and Giancarlo Succi
Centre for Applied Software Engineering
Free University of Bolzano

Quality in FLOSS communities – motivation for the research

- Understanding the perception of trust in FLOSS communities
- Trust in:
  - development processes,
  - methods, approaches,
  - products and development tools.
- The increase of FLOSS development and adoption
- Lack of quality assurance metrics
- Goal - Creation of a CMMI-like methodology

Important FLOSS characteristics

- Features oriented development
- Parallel development, globally distributed development
- Independent peer reviews
- High motivation, high user involvement
- Rapid releases
- Importance of development environments (SourceForge)

Scope of the research

- Analyse of seven largely used FLOSS products such as:
  - Apache HTTP Server
  - Eclipse
  - Emacs
  - Linux kernel
  - Mozilla project
  - GNOME
  - Debian

Methodology used for the research

- We did interviews and case study research when we could not interview people personally
- The first version of the research was qualitative
- Creation of a semi-structured questionnaire
- Source of case study research: web pages, CVS repositories, mailing lists, forums, and others.

The questionnaire

- Based on the Goal Question Metric approach
- Iterative design of the questionnaire
- Goal: Evaluate the adoption of FLOSS in the software industry
- Question: 53 different questions with sub questions
- Metrics: Level of adoption and trust in the quality of the FLOSS development process
- Interview procedure precisely defined
- Anonymity of the interviewees

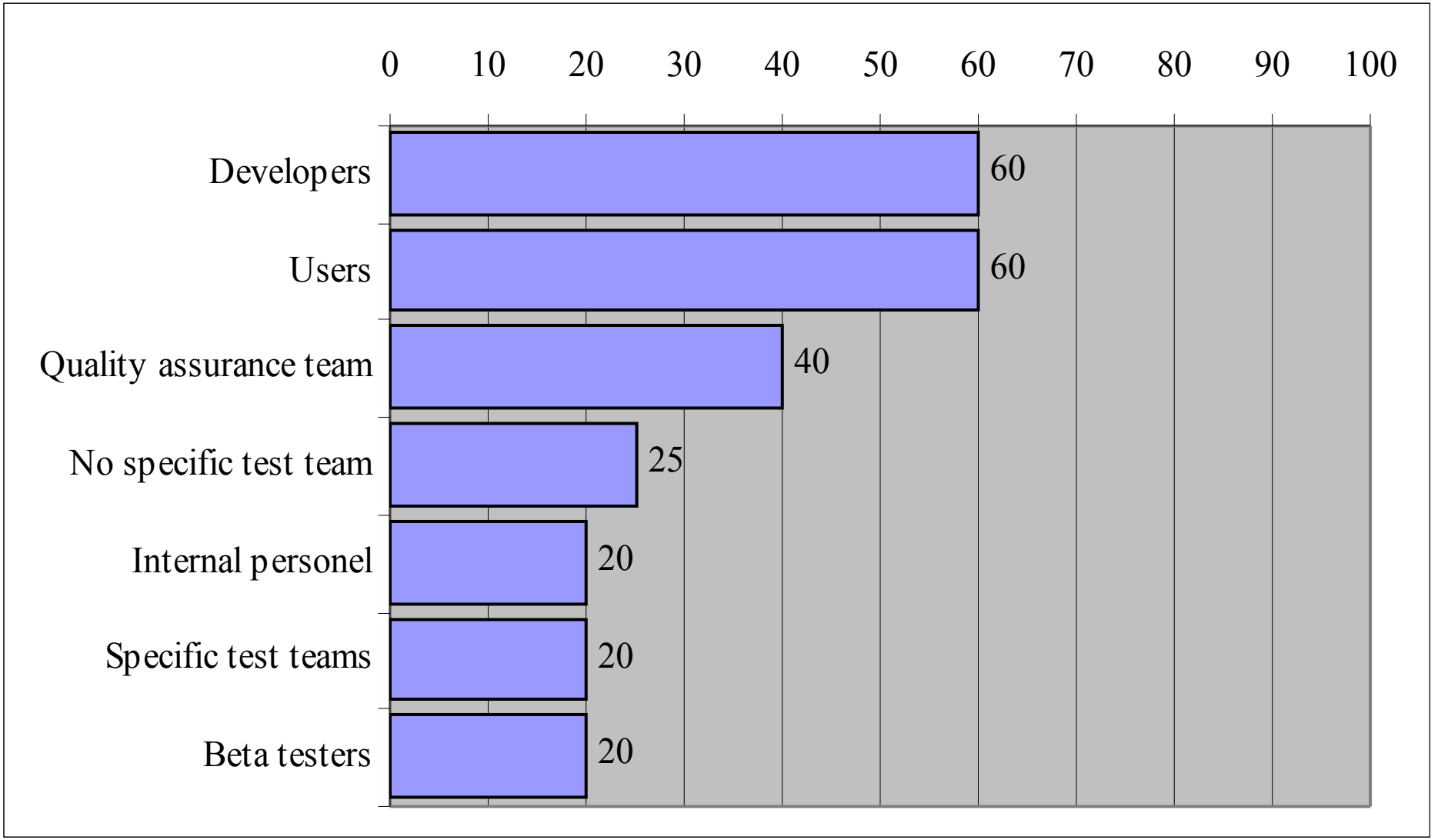
Components of the questionnaire

- Personal information
- Company information
- Role of the organization with respect to FLOSS
- Issues that can be taken into account when deciding whether to adopt FLOSS
- Trust
- Quality assurance
- General questions
- Roles and responsibilities
- Architecture definition
- Development techniques and practices
- Tools used
- Features to implement
- Documentation and bug management
- Version control and people management
- Business model
- Workflows of the processes identified

Trust

- Openness of the whole development process
- Openness of the planing process
- Testing and integration builds
- Availability of intermediate milestones
- Implementation following the milestones
- Communities first try to guarantee classic quality criteria and then additionally they provide FLOSS oriented criteria

Who is testing FLOSS products



Roles, responsibilities and features to be implemented

- Number of developers involved: 25 -1000
- Roles: simple user, developer, committer, Project management committee
- Existence of a foundation
- Good contributions increase privileges
- New features are proposed in bug tracking systems, mailing lists, or forums
- Existence of new features implementation schedule

Technology issues

- The architecture is usually defined incrementally
- Often is following a set of well defined ideas or rules
- Importance of open standards and interoperability
- Availability of a high quality documentation
- Availability of bug/issue tracking systems

Tools used

- The operating system used for development: Linux or other UNIX-like systems
- Adaptation to other operating systems
- The main development language is C/C++ (longevity of projects surveyed)
- Number of different tools used for development is very large; in some cases up to 820 tools
- Most common tools: CVS, Bugzilla, mailing lists

Business issues

- Sources of revenues: supporting services, courses, publications, adaptations of the software, and others.
- Indirect benefits: improved reputation, working in a positive environment, advancing of FLOSS software, possibility to get better jobs, others.
- Support to the FLOSS movement from important software industry players.
- Benefits to companies: publicity, knowledge, attracting young developers

Conclusions

- Key trustworthy elements: number of downloads, the longevity of the project, the level of activity in the community.
- FLOSS communities try to fulfil generic quality requirements and then additional FLOSS requirements
- Quality elements in FLOSS: license used, documentation, testing (based on large user bases)
- Growing involvement of the software industry