PyPy – A Case Study of a F/OSS Community

Alastair Burt (burt@dfki.de)
Beatrice Düring (bea@netg.se)
Holger Krekel (hpk@trillke.net)

http://pypy.org/
http://codespeak.net/pypy

World Domination

- Quote from Linus Torvalds, when he first asked for help in building Linux.

Talk Structure

- Introduction
  - Free / Open Source Software.
  - Python programming language
  - Elements of typical F/OSS development.
- View from the Trenches
  - Typical Python development.
  - PyPy - building a better Python.
  - A F/OSS community and the EU.
- Agile Programming Practices
  - Best practice in software engineering.
  - Agile methods and the typical F/OSS project.
  - Agile methods and PyPy - sprints.
- Discussion

Free and Open Source Software

- Four freedoms.

What is Python?

- Executable pseudo-code. Example.
- Guido van Rossum - Benevolent Dictator for Life (BDFL).
- The typical Pythonista.
- Not the most popular F/OSS language, but has many dedicated fans:
  - Google.
  - Tim Berners-Lee - plane flight coding projects.

Python Principles - What Shapes the Community

- Priorities and principle - Gabriel "Worse is better”
- Python principles - highlight the interesting ones.

F/OSS Community Practices

- Mailing list - no email, no community.
- Source code management - read / write access.
- Bug / feature request tracking.
- Newsgroups, forums - users and developers.
- Web page:
  - About, News, Screenshots
  - Download - often stable and development branches.
  - Support - Documentation, FAQ, Wiki, archives of mailing lists.
  - Related projects.
- IRC - chat.
- Developer weblogs.
- Newsletters.

F/OSS Community Practices - Formal Structure

- Sub-communities in larger projects - special interest groups.
- Regular meetings and conferences.
- Non-proft organisations:
  - Organise meetings and marketing.
  - Hold copyright.
  - Parallel to technical structure.
- Semi-formal decision processes:
  - Python PEPS - proposed changes in the language and development.
  - - BDFLs and the art of saying "no".

Personal Background

- Worked in gaming companies, banks and car companies for several years.
- Studied computer science.
- Left well paid job and went into open-source scenes (2001).
- Various project involvements, started PyPy? 2003 by inviting people to the first "sprint".

What makes Open Source Communities like Python Work: the People Factor

- Collaborative - driven by interest.
- Communication - quite transparent to everyone involved.
- Email / IRC / version-control.
- Organization - rather informal.

Technical Production Factors

- Automated test driven development.
- Specific expertise/special interest.
- Version control (Subversion).
- Releases.

Typical aspects of the Python Community?

- Lively community.
- Lots of different python implementation projects.
- Good contacts between the projects.
- Maybe less fragmented than other OSS communities?

PyPy: the Vision

- Founders came from the Python community.
- Sprints were the inital factor.
- What is PyPy / Python - one of the five most used programming languages today.
- Grass root approach.

OSS and EU funding: PyPy? as a Case Study

- Driven by partially EU funded and non-EU funded parties.
- Focus on avoiding friction and turning PyPy? into a long term project.
- IBM or Sun have done similarly challenging projects in more time and with more funding.
- Yet not found completely satisfying "funding" interactions with communities.

PyPy Technical Status

- Three public releases in 2005, well received by the community.
- Core deliverables fulfilled.
- Contributors add different directions.

PyPy: It's all about Communication

- Pypy-sync meetings, 30 minutes IRC .
- Day-to-day IRC discussions.
- "This week in PyPy".
- Mailing lists: pypy-svn/eu-tracking tracks code and document changes.
- Around 20000 visitors per month on website.
- Lots of blogs and pypy-dev (developer/researcher list).
- 300-500 people across the world following the project.

All Good and Well but the Main thing are: Sprints

- One-week intense work-meetings with one break day.
- EU and non-EU researchers/developers get together.
- Daily planning sessions.
- Pair programming.
- Evolving and adapting to more attendants.
- Organisational/management tasks happen also on sprints.

Next

- Tackling research and technical goals (challenging!).
- Mid-term EU review planned for 20th january.
- Looking into adjusting some work planning.
- Increased dissemination, attending conferences (movie features?).
- Start talking to and interact with commercial stakeholders.

Agile Development

- Need to handle rapid change in commercial software development.
- How do agile approaches fit distributed, open-source projects?

Core of Agile Practices: the People Factor

Agile processes are designed to capitalize on each individual and each team´s unique strengths (Cockburn, Highsmith, 2001).

- OSS nature of teams: self-organized, intensely collaborative - fit the agile approach.
- OSS teams are an unique implementation of agile practices.

Agile Approaches

- Aim at:
  - *Reducing* "cost of information",distance from decision-making
  - *By* physical location, unorthodox dissemination.
  - *Resulting* in improved sense of community, team "morale"

Agile Teams

Agile teams are characterized by self-organization and intense collaboration, within and across organizational boundaries

(Cockburn, Highsmith, 2001)

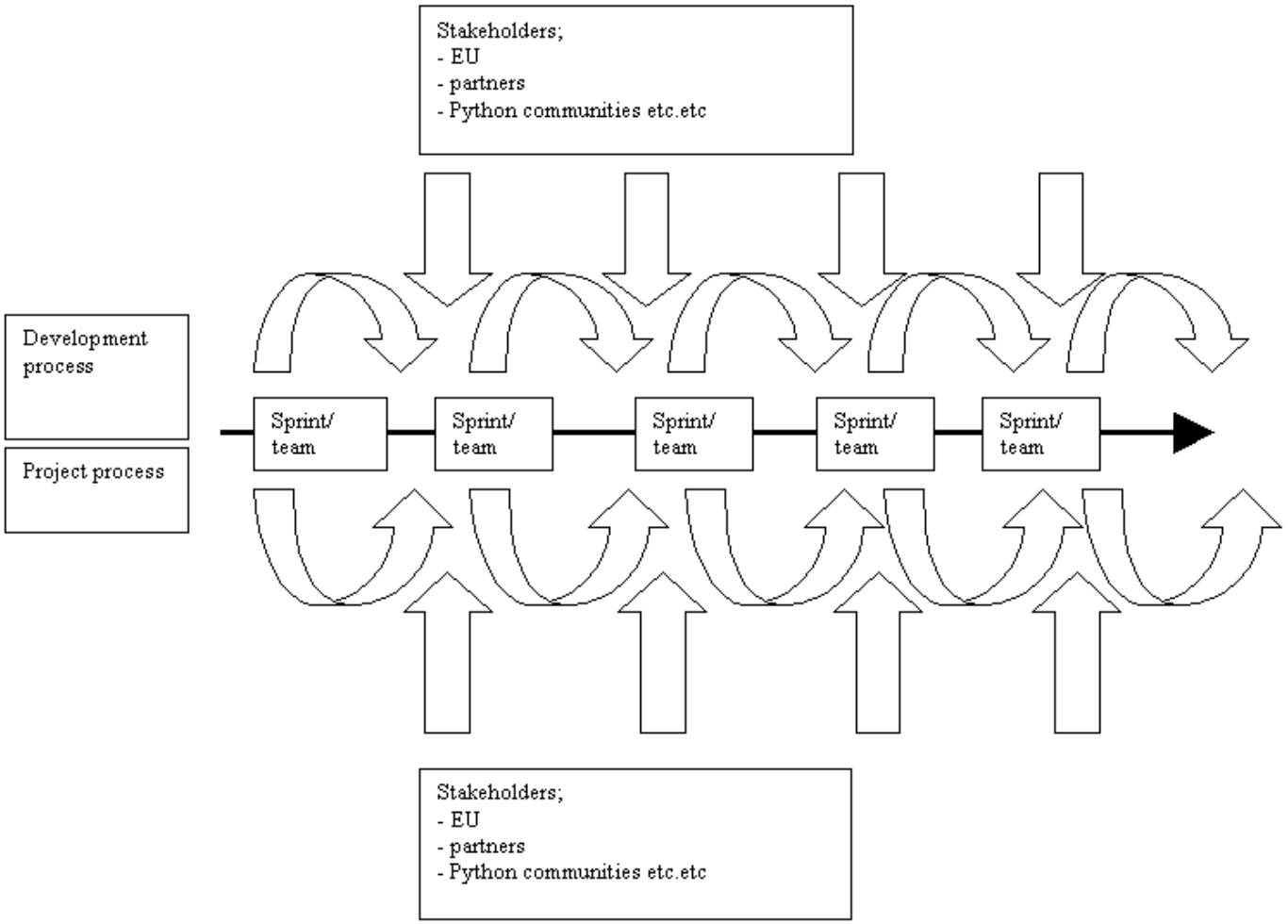
PyPy

- How do one structure an agile OSS community into a consortium of 7 partners?
  - Create developer driven organization.
  - Roles and responsibility (management team + tehnical board).
  - Uses IRC channels, version control (SVN)on consortium level.

Sprints

- Sprinting is central to the PyPy project:
  - Funded as well as non-funded work.
  - Dissemination (talks, tutorials, pairprogramming).
  - Consortium activities (meetings, planning, coordinating wp work).
  - Contribution from community via "physical persons" structure.

Sprint Process



Evolution of PyPy Sprints

- Evaluations are done with "external" participants.
- Different forms of sprints with different focus.
- Sprints at conferences are growing into workshops.

Results of the Sprints

- More people knowing and understanding the vision of PyPy.
- More people "recruited" into the community.
- People "recruited" into the consortium (physical persons).
- People "recruited" into PyPy companies.

Discussion

- In what way has the Internet enabled new forms of peer production? Can this be extended beyond software development?
- How far has the informal, anarchic world of F/OSS development influenced software engineering and vice versa?
- How can the F/OSS communities help the EU achieve its goals?
- How could the EU potentiate F/OSS development?