PyPy – A Case Study of a F/OSS Community

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

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

World Domination

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs.It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-hard disks, as that's all I have :-(

Talk Structure

- Introduction
  - Free / Open Source Software.
  - Python.
  - 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 a nd PyPy - sprints.
- Discussion

Part I

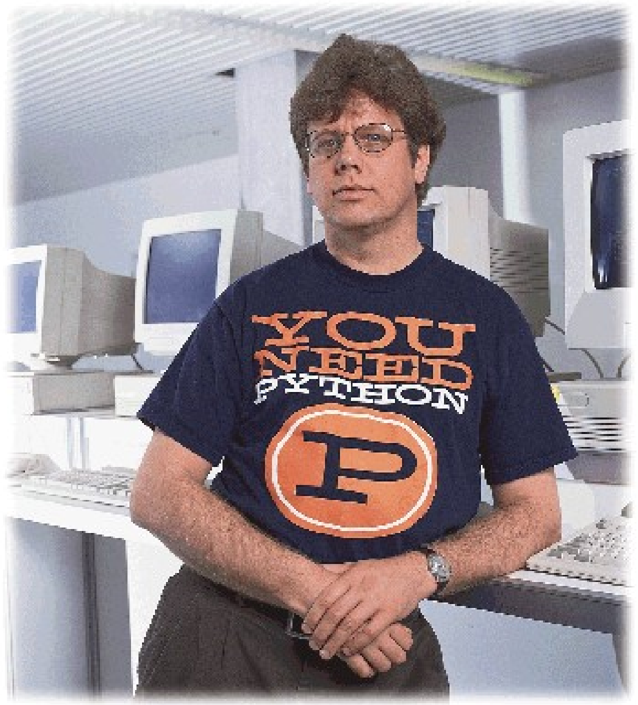
Introduction

Free and Open Source Software

1. The freedom to run the program, for any purpose.
2. The freedom to study how the program works, and adapt it to your needs.
3. The freedom to redistribute copies so you can help your neighbour.
4. The freedom to improve the program, and release your improvements to the public, so that the whole community benefits.

What is Python?

- Executable pseudo-code.
- Guido van Rossum - Benevolent Dictator for Life (BDFL).



What is Python?

- 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

- **Beautiful is better than ugly.**
- **Explicit is better than implicit.**
- Simple is better than complex.
- Complex is better than complicated.
- **Flat is better than nested.**
- Sparse is better than dense.
- **Readability counts.**
- Special cases aren't special enough to break the rules.
- Although practicality beats purity.
- Errors should never pass silently.
- Unless explicitly silenced.
- In the face of ambiguity, refuse the temptation to guess now.

Python Principles - What Shapes the Community

- If the implementation is hard to explain, it's a bad idea.
- **There should be one -- and preferably only one -- obvious way to do it.**
- Although that way may not be obvious at first unless you're Dutch.
- Now is better than never.
- Although never is often better than right
- If the implementation is easy to explain, it may be a good idea.
- Namespaces are one honking great idea -- let's do more of those!

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.
- IRC – chat.
- Newsletters.
- Web page:
  - About, News, Screenshots
  - Download - often stable and development branches.
  - Support - Documentation, FAQ, Wiki, archives of mailing lists.
  - Related projects.
- Developer weblogs.

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".

Part II

View from the Trenches

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 F/OSS Communities 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.

Sprints

- One-week intense work-meetings with one break day.

Sprints

- 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.

Part III

Agile Development

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)

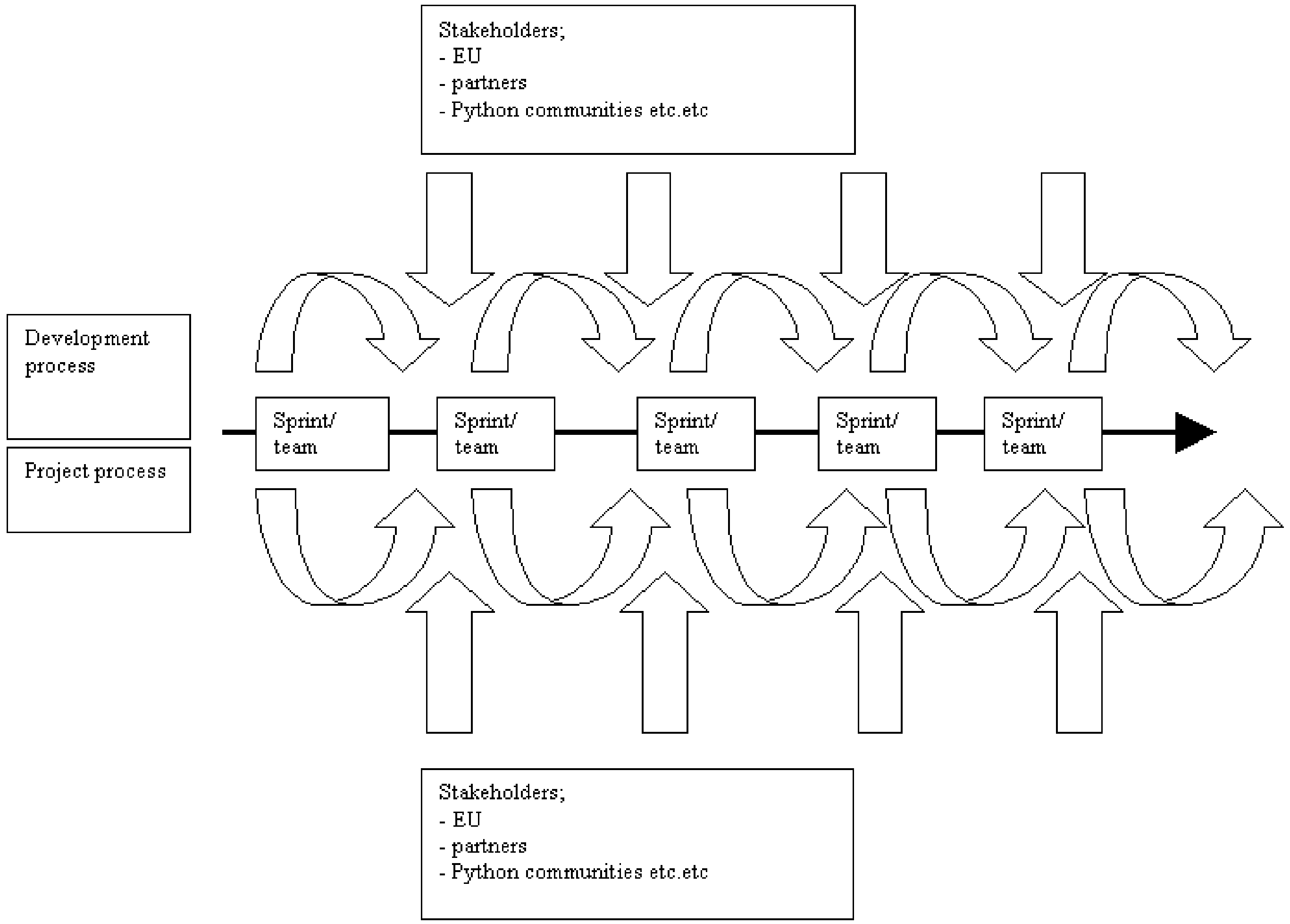
- How do one structure an agile OSS community into a consortium of 7 partners?
  - Create developer driven organization.
  - Roles and responsibility (management team + tehnical board).
  - Uses IRC channels, version control (SVN)on consortium level.

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.

Sprints

Sprint Process



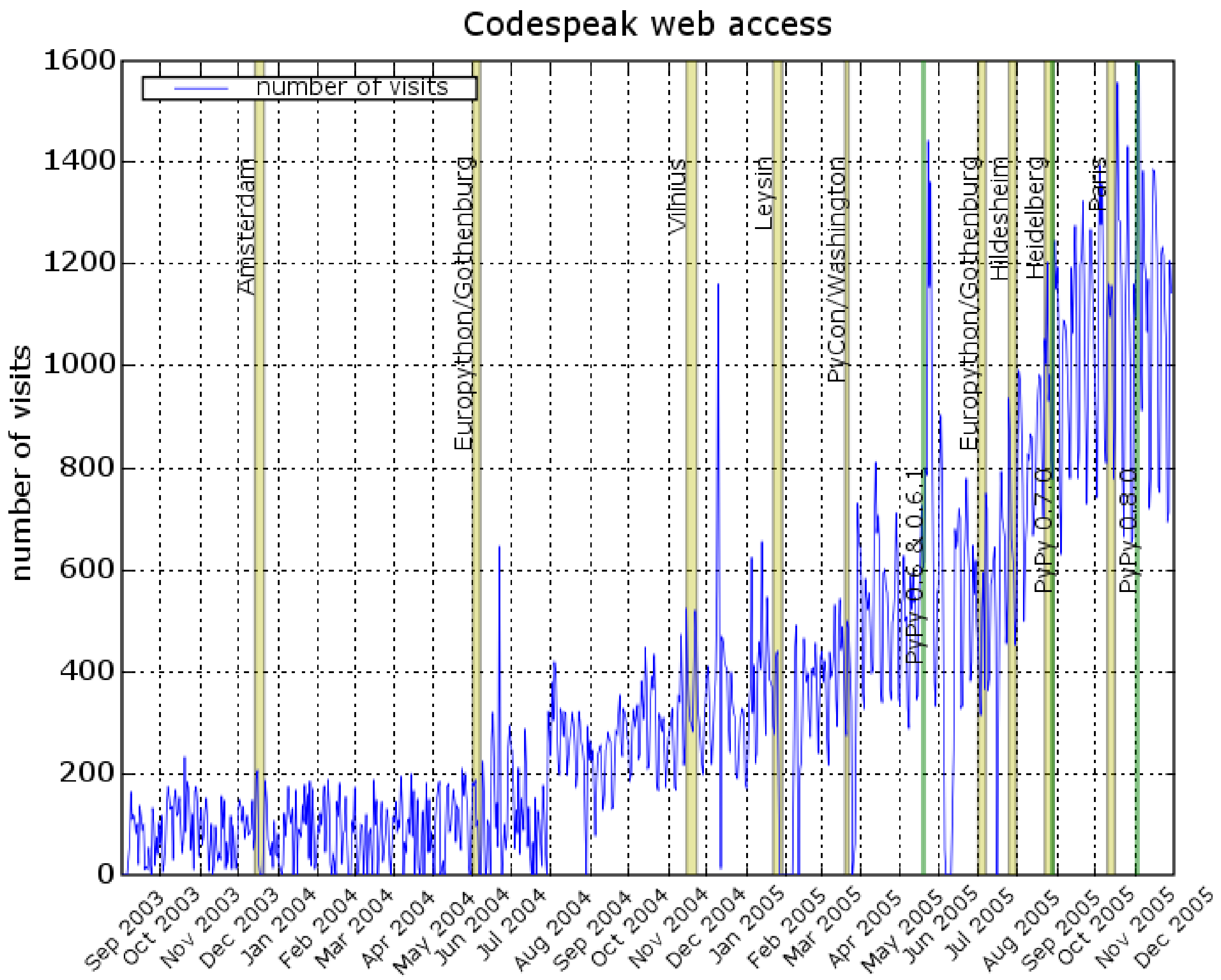
Evolution of PyPy Sprints

- 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.

The Effects of Sprints



Part IV

Discussion

Discussion

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

Discussion

Your questions?