Agile Open-Source Methods, Businesses and EU-funding

Author: Bea During (Change Maker)
Author: Holger Krekel (merlinux)
Date: 26th February 2006, PyCon, Dallas

1 What is PyPy?

- Next generation Python implementation
- Grass-root open source effort
- A partially funded research project
- Technical aims: flexibility and speed
- Method aims: evolve sprint-driven development

2 Bootstrapping PyPy (2003)

- Initial sprint with Armin Rigo, Christian Tismer, Holger Krekel, Michael Hudson and then Samuele Pedroni ...
- Driven by psyco/stackless experiences
- "Minimal Python" inital project name
- Focus on correctness of concepts, then speed

3 PyPy test-driven development

- Identify problems/evolution by tests first
- Our own testing and development tools (py.test)
- Rule: first get the semantics and concepts right! optimize later!
- Today around 3000 tests (plus CPython regression tests)

4 PyPy is (not) a funded EU project

- PyPy wasn't a funded project
- PyPy isn't a funded project
- PyPy is a funded project
- Conclusion: it's more complicated

5 EU project consortium

- AB Strakt (Sweden)
- merlinux (Germany)
- Change Maker (Sweden)
- Heinrich Heine University (Germany)
- DFKI (Germany)
- Tismerysoft (Germany)
- Logilab (France)
- Impara (Germany)

6 EU contractual work structure

- "Description of Work" for two years
- 14 workpackages and 58 deliverables, 3 phases
- Sprints every 6th week (coordinating development and management work)
- EU project aspects enforced mid-term/long-term focus
- not a usual open-source approach
- **good and bad**

7 Balance of interests

- PyPy was first and still is a network of people
- ...but EU only funds organisations
- Developers drive the technical project
- Companies co-finance 50% of all costs
- EU wants challenging research goals and tracking of goals
- **it is all about finding good models for co-operation**

8 Current developer work structure

- Weekly 30 minute synchronisation meetings
- Open collaborative open-source work style
- Representation through Trusted "Technical Board" within the EU project
- Research/architecture informally guided by accepted experts

9 Agile practises: the people factor

- "Agile processes are designed to capitalize on each individual and each team's unique strenghts" (Cockburn, Highsmith, 2001)
- OSS nature of teams: self-organized, intensely collaborative - fit the agile approach
- OSS teams are an unique implementation of agile practices - why?

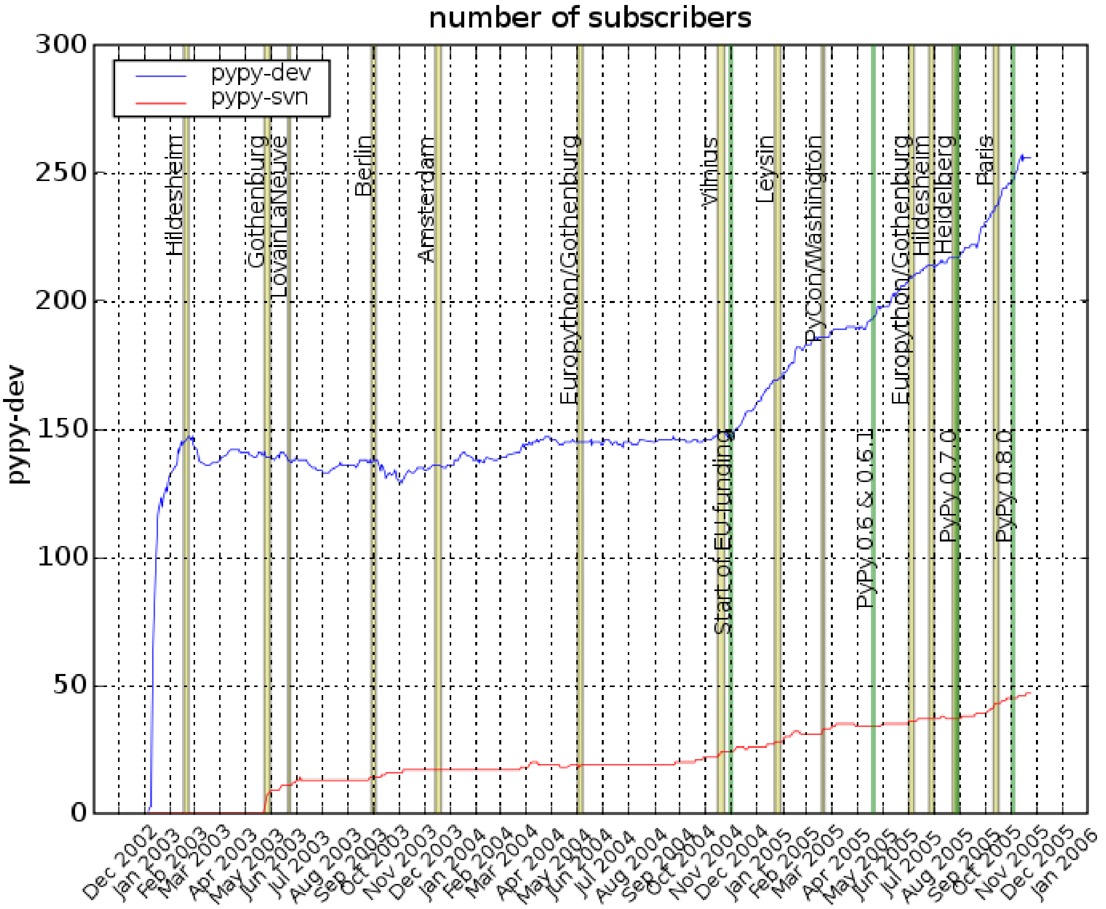
10 Origins of sprinting

- Scrum (Agile community): 1 month long iteration of development work, increments (also supporting activities: planning, documentation, tracking work, evaluation)
- Zope Foundation (Python Community): "two-day or three-day focused development session, in which developers pair off together in a room and focus on building a particular subsystem".

11 Sprinting the PyPy way

- First-contact sprints e.g. around conferences
- Core sprints for reaching milestones, releases
- 7 days with 1 break day, every 6th week
- Typical activities: daily planning/status meetings, closure meetings, sprint reports, pair-group programming, tutorials, EU consortium issues
- Rotating moderation/organisation

12 Sprints facilitate participation



13 Managing diversities

- **Developer driven process and formal project organization**
  - management team, technical board and partners
  - sprint organising
  - planning and focusing on technical tasks
- Constant risk of added workload of management work on core developers

14 More managing diversities

- **Agile strategies and Formal EU requirements**
  - written high level requirements
  - change control structures complicated
- Constant risk of missing opportunities and not creating/reacting to change fast enough

15 Even more diversities ...

- **OSS community and hierarchies for "conceptual integrity"**
  - pypy-dev/core developers in technical board
  - industrial usage vs research oriented work
- Risk for unbalancing the community

16 Hitchikers guide ...

17 Learnings

- Share a challenging vision!
- Respect and utilize strengths of different cultures and people involved

- Design minimalistic project structures
- Learn as a group and create changes, not just react to change

18 Outlook on whole project level

- EU project to finish November 2006
- Improve interactions with community & contribution
- Exploring Commercial opportunities ... hiring opportunities ...
- Taking care about post-EU development (2007++)
- Sprints: Lovain La Neuve, Tokyo, EuroPython, Ireland
- http://codespeak.net/pypy and http://pypy.org