Open Source, EU Funding and Agile Methods

Author: Bea During (Change Maker)
Author: Holger Krekel (merlinux)
Date: 29th December 2005

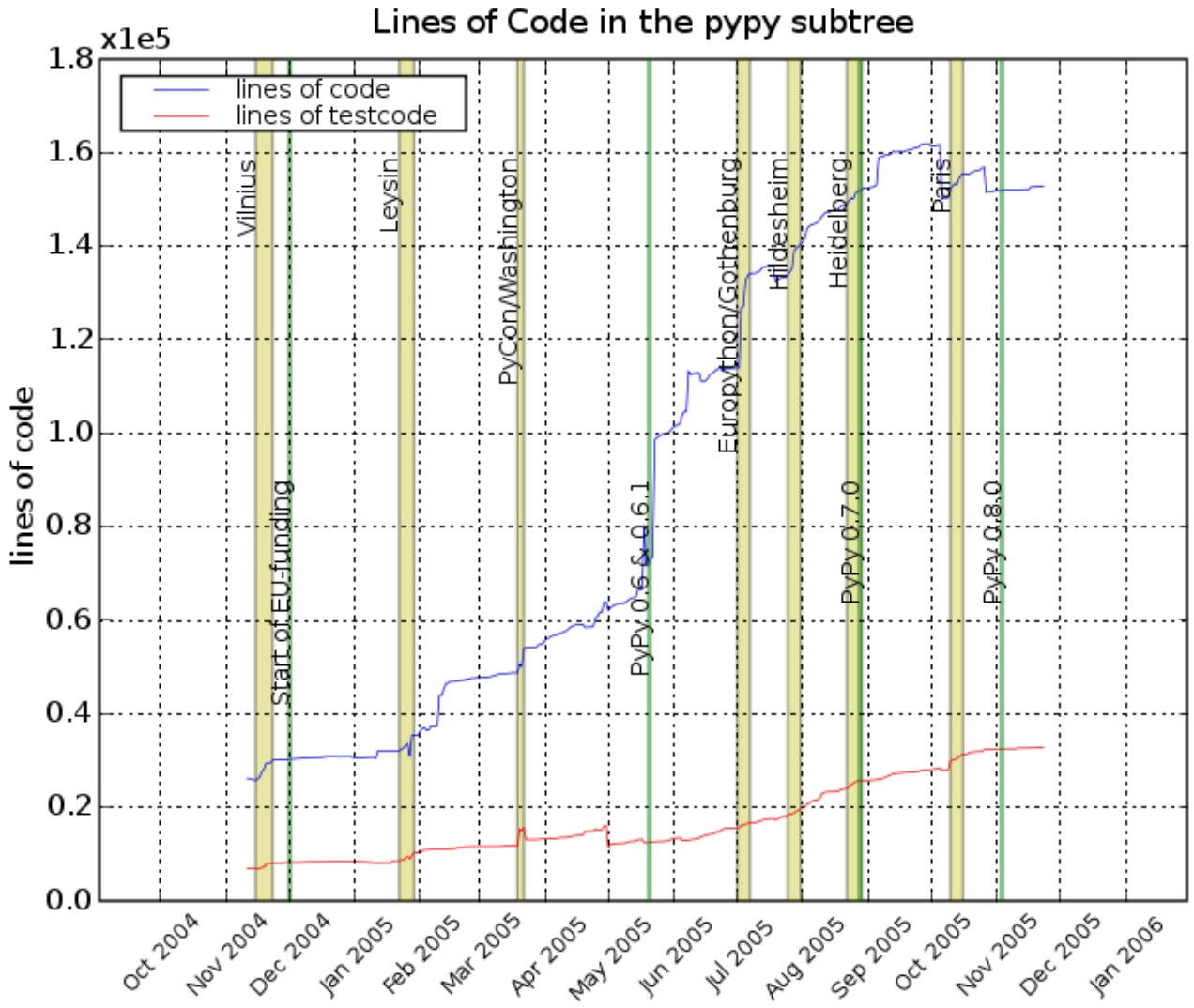
1 How it got started

- 2003 first emails between Armin Rigo, Christian Tismer and Holger Krekel
- participated in zope3 coding events ("sprints")
- initial invitation for a one-week sprint to Trillke, Hildesheim
- participants got to know each other at conferences
- goal: Python implementation in Python (various motivations)

2 Key elements of the technical development

- test-driven from the start
- driven by architectural experiences
- welcomed by python community
- based on zen of python / python culture
- focus on correctness of concepts, then speed
- evolutionary step by step development

3 Lines of Code and tests



4 Python community

- strong open-source cultural background
- strong focus on glue & integration esp. with C/other languages
- few flaming fights inside / much collaboration
- has many windows and unix hackers
- CPython: main Python version (BDFL'ed by Guido)
- Jython: top scripting langauge for Java

- IronPython (MS): compiles to .NET's CLR
- PyPy: self-contained - self-translating - flexible

5 Evolving agility

- all large python projects rely and depend on automated testing
- several projects regularly "sprint" and work together distributedly
- community conference EuroPython in 2002 (now yearly)
- many test tools and methods available
- next EuroPython 2006: June at CERN (Geneva)
- ... with dedicated agile development track

6 PyPy test-driven development

- identify problems/evolution by tests first
- our own testing and development tools
- rule: first get the semantics and concepts right! optimize later!
- today around 3000 tests (plus CPython regression tests)

7 PyPy's evolution as a project

- 2003: four one-week meetings, basic architecture evolved
- mid 2003: realisation that we'd need to work full time on it to make it succeed
- idea for EU funding was born!
- collaborative work on a proposal ...

8 EU funding

- Proposal written during sprints as well as distributed (submitted Oct 2003)
- got good reviews from EU contracted experts
- negotiation and finalisation: 1 year!
- 2 year project, 1.3 Million Euro funding
- contractual framework + reporting obligations

9 Work organisation (in the EU proposal)

- 14 workpackages and 58 deliverables, 3 phases
- Need for consortium meetings every month (IRC)
- Sprints every 6th week (coordinating development and management work)
- EU project aspects helped to gain mid-term/long-term focus

10 Balance of interests

- developers want to (continue to) drive the project
- companies have to co-finance 50% of all costs (travel/salary), commercial interests
- EU wants challenging research goals and tracking of goals
- **at all levels it is about finding models for co-operation that fit**

11 Sidenote: Free co-operation basics

- Christoph Spehr's "foundation of free co-operation":
- negotiate any model you want (including dictatorship)
- question and change rules and roles at any time
- everyone can "leave" the co-operation without penalty
- leaving party can even take a share with him/her

12 Developers collaboration

- 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

13 Company collaboration

- contractually through the EU consortium
- exchange of knowledge and people, shared tools
- evolving commercial opportunities US companies asking for consulting (test tool) or wanting to hire/pay developers on pypy related tools

14 Organising the consortium

- PyPy was first and still is a network of people
- but EU only funds organisations
- 7 partners, 3 previously not involved in the PyPy community
- 6 partners only partially funded (50% cost models)
- 2 new companies: "forced" entrepreneurship

15 Consortium Meetings ...

16 Core of 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?

17 Agile approaches aim at ...

- reducing ... "cost of information",distance from decision-making
- by ... physical location, unorthodox exchange of knowledge
- resulting in ... improved sense of community, team "morale"

18 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".

19 PyPy sprints

- The project "started" with a sprint
- Changing facilities and location as a strategy (Vilnius, Lovain LeNeuve, Leysin, Gothenburg, Paris, Heidelberg, Hildesheim, Washington etc)
- The nature of sprints have evolved since the project started 2003 and since recieving partial EU-funding 2004/2005

20 Sprinting the PyPy way 1

- Planning: location, venue, rough goals and activities, preparation with local hosts

21 Sprinting the PyPy way 2

- Doing: start up meeting, daily status meetings, pairprogramming

22 Sprinting the PyPy way 3

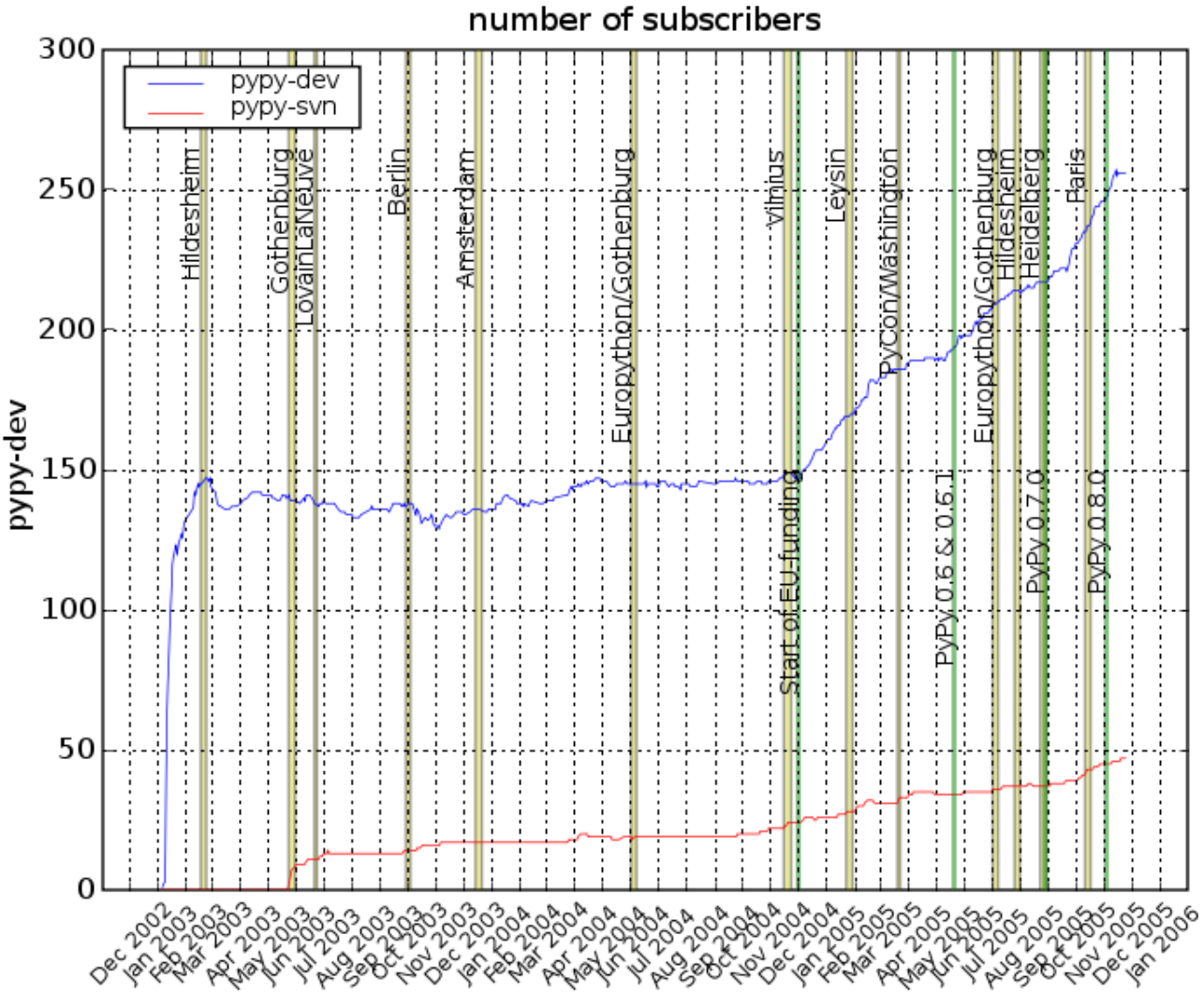
- Closing: closure meeting (planning work between sprints), sprint reports, evaluations

23 Sprinting the PyPy way 4

- 7 days with 1 break day
- "open" sprints and "closed" sprints - levels of PyPy knowledge in participants

- sprints at conferences (PyCon, EuroPython)

24 Effects of sprints on community participation



25 The different cultures of the PyPy project

- OSS/Python culture (agile and distributed workstyle)
- EU project culture
- Traditional project management culture
- Chaospilot (actionlearning and process design) culture
- 5+X different national cultures

26 The challenge: managing diversities part 1

- **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

27 The challenge: managing diversities part 2

- **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

28 The challenge: managing diversities part 3

- **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

29 Hitchikers guide ...

30 Problems and perspectives

- **confrontation with people from**
  - same planet, different place
  - different planet
  - different solar system
- what follows is slightly abstract ...

31 Working with people from the same planet

- generally shared perspectives, synchronised rotation around a common center ... on project success

- different continents can be hard enough
- potential for misunderstanding/friction

32 Different planet

- shared view regarding the center (project success)
- quite different working perspectives / methods (e.g. "open-source collaborative" versus "formal traditional")
- potential for mistrust and dis-connection

33 Different solar system

- missing shared focus on project success?!
- alien or alienating interests
- potential for defense/attack thinking, secret agendas

34 The universal truth is ...

- often good intentions (!) even from aliens
- confrontation with lots of levels, planets and solar systems
- the challenge is to find a fitting model for case-by-case co-operation!
- identify location in universe and try to match and synchronize

35 Cross-project pollination

- zope-europe, canonical, Calibre
- dissemination: universities, IONA, Intel, HP ...
- Alan Kay
- Squeak (21c3)
- ...

36 Conclusion / Food for thought

- A shared and challenging vision
- Respecting and "exploiting" strengths of the different cultures involved
- Designing minimalistic project structures channeling work, not hindering work
- Room for group learning and creating change - not just reacting to change

37 Outlook on whole project level

- surviving the EU review in Bruxelles 20th January 2006
- improve interactions with community & contribution
- taking care about post-EU development (2007++)
- visiting Mallorca, Texas, Tokyo, Ireland, ...
- commercial opportunities ... hiring opportunities ...
- *Questions?* (talk to us ...)

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