IST FP6-004779

PYPY

Researching a Highly Flexible and Modular Language Platform and Implementing it by Leveraging the Open Source Python Language and Community

STREP

IST Priority 2

D01.2-4 Overview on Project Organization Activities

Due date of deliverable: March 2007

Actual Submission date: March 12, 2007

Start date of Project: 1st December 2004   **Duration: 28 months**

Lead Contractor of this WP: DFKI

Authors: Beatrice Düring (Change Maker) Lene Wagner (merlinux)

<div align="right">

Revision: Interim

</div>

Dissemination Level: PU (Public)

Revision History

Date	Name	Reason of Change
2007-02-13	Lene Wagner	report sceleton packaging D01.2-4
2007-02-25	Beatrice Düring	adding data to all sections
2007-02-26	Düring/Wagner	amendments to all sections
2007-02-27	Stephan Busemann	internal review
2007-03-11	Lene Wagner	streamlining, amendments to exec summary/intro
2007-03-12	Lene Wagner	outlook part, new title
2007-03-12	Beatrice Düring	internal review of last changes
2007-03-12	Carl Friedrich Bolz	Publish Interim on the Web Page

Abstract

In this Technical Report we present the reporting procedures put into place in the PyPy project, including an overview of the deliverable reports submitted during the EU project. We also present the management structure and the regular meeting procedures we employed for coordinating work and decision making processes. Sprint-driven development has been a core practice in the project and sprints were the main physical meeting form of the developers (both Consortium and community). We list here the sprints arranged. Finally, we describe the information infrastructure (mailinglists, website, newsletters and IRC channels) that we used for coordinating the PyPy project.

Contents

1 Executive Summary

The conjunction of a distributed work environment with team members spread all over Europe working in very close collaboration on a day-to-day basis formed a challenge for coordinating the PyPy EU project. An additional requirement, underlying the coordination activities, was the aim to conduct a transparent and equal style of communication and decision making on all involved levels.

This setting required the evolution of sophisticated techniques of virtual communication, both for the organisation of technical coordination and mentoring and the formal organisation of the Consortium itself.

Similar to the project's style of work, report writing was a shared effort of numerous persons, requiring centralized monitoring in the form of "report release management". To track progress of both work and cost/person-month consumption in this dynamic environment, a number of internal reporting methods were employed.

During the EU funding period, various coordinating bodies such as the Consortium, the Technical Board, and the Management Team were put into place all interacting as needed with the active developer group consisting of EU-project participants and contributors from the Community. The planning and coordinating of sprints proved a major task that traversed all organizational levels of the project.

In order to organize the information flow on the different project levels, a set of virtual information channels was used, including mailing lists, IRC (Internet Relay Chat) channels and the project's website, in general aiming at providing an infrastructure that is easy to use, transparent and open to external parties.

These activities, involving both technical coordination and formal Consortium management, have obviously not only been carried out by the management team itself, but also by those who were directly involved with the matters at hand.

After the end of the EU project period, the organizational structure of the project is likely going to change in many ways. While, as described in the D02.1 (2.1) report, the development and communication infrastructure will continue to be available and maintained, many of the EU-project specific organizational levels will probably not be used any more, such as the Consortium, the Technical Board or the Management Team in their current set-up.

Current plans, being discussed among the Technical Board, community members and Consortium partners outline a continued open project structure, allowing all interested parties to participate in the known communication and contribution channels. There will likely be a trusted fixed group of PyPy core contributors that is going to take care for codebase consistency, development, release and sprint planning. This group is also meant to take care of negotiating with external parties about funding the overall project progress and sprints in particular. It thus acts as an interface towards both the community and commercially interested partners, and will also be a mediating instance in case of conflicts. According plans continue to evolve and will be described separately, e.g. in the Periodic and Final Activity Report (PER) (FIN).

2 Introduction

2.1 Purpose of this Document

This Technical Report describes the management activities done and infrastructures established to coordinate the project both on a technical and on a formal level. This report tries to

facilitate an understanding of the organization of PyPy's distributed work environment.

2.2   Scope of this Document

This document presents an overview of the coordination means that have been of use in the project and core results of the coordination efforts, such as reports collected and meetings held. It does not give a detailed insight into how the Consortium has been structured and has evolved during the project (relating to partners, contractual amendments or budget shifts) - nor does it present any technical results of the development process. As such this document is a complement to the Periodic and Final Activity Reports.

This Report contains the information of the three Technical Deliverables D01.2, D01.3, D01.4, packaged into one document.

2.3   Related Documents

For more in-depth understanding of the context and evolution of the project please refer to the following documents:

- D14.1 *Report About Milestone Phase 1*
- D14.3 *Report About Milestone Phase 2*
- D14.4 *Report About Milestone Phase 3* (to be submitted April 2007)

The technical infrastructure providing for the on-line coordination activities is described in:

- D02.1 *Development Tools and Website*

3   Collecting and Monitoring of Reports Including Reports to the EU

3.1   Report writing procedure

When we discuss reporting in PyPy, we are referring to both the process of writing deliverable reports of the EU contract and the formal and informal reporting procedures within the Consortium that were put into place to track the developments of the project in a transparent way.

3.2   EU reports

EU Reports were usually written by more than one person and were also reviewed by other persons involved in carrying out the related tasks and, whenever feasible, by external stakeholders with deep knowledge of the domain area.

The organization of writing reports, monitoring of deadlines and review procedures was done by a commonly assigned 'report release manager' - a technique borrowed from open source

software release management. The responsibility of actually delivering correctly and in time remained with the involved partners in the work package, especially the lead partner.

For a convenient overview on the actual delivery dates of the Contractual and Technical Reports, we intend to include a table listing those Dates in the final version of this report.

3.3 Internal reporting

Internal technical reporting took place on partly formal, but more important informal levels. A central reporting tool have been the automated svn-commit-notication emails, providing all involved developers with a traceable view on each change to the shared codebase and documentation, including the author's name and a short log message describing the implemented change.[1]

Day-to-day technical assessment was carried out by frequent discussions of the people involved with carrying out the work (via IRC, Internet Relay Chat, mainly, on the channel #pypy at freenode.net), stating problems and commonly implementing solutions without formal internal reporting procedures. At the end of the First Reporting Period and for the 4th and 5th contract amendment, however, the Technical Board collected status reports of the single workpackages from all partners.

Time and cost tracking was done on a central level. All partners committed their timetrack data monthly to a shared repository, providing time consumption transparency to the management team and all other Consortium partners. In addition to the cost statements of the First Reporting period, two interim cost estimates (draft form c information) were collected in summer 2005 and 2006 in order to closely track budget consumption and adjust the budget distribution accordingly.

4 Organize Meetings and Sprints

4.1 Sprints

Sprints have been the core development coordination practice used in the PyPy EU-project. As such sprint planning and organizing as well as having the possibilities to coordinate results working face-to-face when co-located meant that sprints meant coordinating not only technical work but also Consortium and management level of work, traversing all levels of the PyPy EU-project.

Sprints[2] have been announced on the PyPy community website, by emailed announcements to the pypy-dev mailinglist as well as by announcing on Python mailinglists such as comp.lang.python. The results of the sprints were documented in sprint reports (see section "Newsletter for External Stakeholders").

Here is a list of all sprints arranged during the project:

Before EU-funding period:

- Hildesheim, Germany (2003-02-17,2003-02-23)

---

[1]The technical implementation of the shared repository and commit-notification structure is described in the deliverable report D02.1 *Development Tools and Website* (2.1).

[2]For a more in-depth explanation of the practices of sprinting done in PyPy, please read report D14.5 *Documentation of the development process* (14.5).

- Gothenburg, Sweden (2003-05-24,2003-05-31)
- LovainLaNeuve, Belgium (2003-06-21,2003-06-24)
- Berlin, Germany (2003-09-29,2003-10-04)
- Amsterdam, The Netherlands (2003-12-14,2003-12-21)
- Europython/Gothenburg, Sweden (2004-06-01,2004-06-07)
- Vilnius, Lithuania (2004-11-15,2004-11-23)

EU-funded period:
- Leysin, Switzerland (2005-01-22,2005-01-29) 14 participants
- PyCon/Washington, USA (2005-03-19,2005-03-22) 13 participants
- Europython/Gothenburg, Sweden (2005-07-01,2005-07-07) 20 participants
- Hildesheim, Germany (2005-07-25,2005-07-31) 20 participants
- Heidelberg, Germany (2005-08-22,2005-08-29) 6 participants
- Paris, France (2005-10-10,2005-10-16)14 participants
- Gothenburg, Sweden (2005-12-05,2005-12-11) 22 participants
- Mallorca, Spain (2006-01-23,2006-01-29) 15 participants
- PyCon/Dallas, USA (2006-02-27,2006-03-02) 20 participants
- LouvainLaNeuve, Belgium (2006-03-06,2006-03-10) 9 participants
- Leysin, Switzerland () 10 participants
- Tokyo, Japan (2006-04-23,2006-04-29) 12 participants
- Düsseldorf, Germany (2006-06-02,2006-06-09) 12 participants
- Europython/Geneva, Switzerland (2006-07-06,2006-07-09) 24 participants
- Limerick, Ireland (2006-08-21,2006-08-27) 7 participants
- Düsseldorf, Germany (2006-10-30,2006-11-05) 14 participants
- Leysin, Switzerland (2007-01-08,2007-01-14) 17 participants
- Hildesheim, Germany (2007-03-01,2007-03-05) 19 participants

4.2 Meetings of the Technical Board

The Technical Board was constituted to allow the core developers of PyPy to coordinate the development work of Consortium partners, ensuring the contractual implementation of the project against the background of the open and partly non funded development process.

The Technical Board was responsible for high-level technical decisions, such as technical priorities, work distribution and dealing with problems affecting the project as a whole, such as the health incident of a core developer in Summer 2006.

It should be noted that none of the core developers represented partners with a budget under the Type of Activity "Management of the Consortium". The role and work of the Technical Board was an integral part in succeeding with the strategy of "leveraging the community". Much decision power over the technical coordination of the partners, all related to work of a management nature, was delegated to the Technical Board.

During the EU project the Technical Board met 15 times, mostly via IRC on the non-public #pypy-tb channel on freenode.net. When deemed necessary the Technical Board invited non-members involved in the issues on the agenda to participate.

4.3   Meetings of the PyPy community: pypy-sync

After the first period of the project, synchronization meetings (`pypy-sync`) were constituted in order to better synchronize work between sprints when working in this distributed and dispersed manner. The strategy of the sync-meetings was to coordinate work of the community. This was non-funded work as well as funded - although the main work coordinated in many cases covered the EU-funded part of the work. Sync-meetings have been open for anyone to attend, and because of that sync-meetings have only been arranged via IRC, on #pypy-sync on freenode.net. For more information on how pypy-sync meetings have been organized in the PyPy project, please read the experience report published for the Agile 2006 conference: "Trouble in Paradise: the Agile Open Source Project PyPy, EU-funding and Agile Practices" (AGILE).

During the EU-funded part of the project 27 sync-meetings have been organized.

4.4   Meetings of the Management Team

The management team held seven meetings with a prepared agenda and according minutes, two of which were held before, five after the implementation of the 'Agile management team' concept (PER). Additionally, informal phone conferences and other remote synchronizations took place, coordinating day-to-day management issues.

4.5   Meetings of the PyPy Consortium

For Consortium related decisions the representatives of the partners gathered for Consortium meetings, mostly held remotely on IRC on the non-public channel #pypy-Consortium, but also physically in relation to project sprints.

During the EU-funded project period, the Consortium held:

- a Kick off meeting in Saarbrücken, Germany (at DFKI), setting up the Consortium Agreement
- 17 Consortium meetings (3 of which in physical form)

5   Project Website / Newsletter for External Stakeholders

5.1   The PyPy website and repositories

The PyPy project used two websites. The first is http://pypy.org, a project website presenting a summary of the objectives and partners involved of the EU-funded part of the project. This page links to the main community website at http://codespeak.net/pypy.

The strategy of the summary project page was to have information through an easy URL for external stakeholders that would not necessarily be too interested in the development side of the project.

The main website has the purpose of catering to visitors that are interested in the development part of the project. There is a clear strategy of avoiding EU jargon in this public community

area. Here news articles, sprint announcements, release announcements, sprint reports and extensive documentation is published.[3]

All content of the web pages is stored in version control system repositories, just like the project's source code. One part of the repositories is closed to the public, namely the Consortium administration section. Here all minutes of Technical Board meetings, Management Team meetings and Consortium meetings are archived, as well as all contractually related documents, cost statements and timesheets from the Consortium partners.

All partners, managers as well as developers working in the partners' organisations, can access all data in this repository. This means that the different partners can view each other's submitted information. This was done with the sole purpose of having as much openness and transparancy within the Consortium structure as possible.

5.2   Newsletter for External Stakeholders

Because of the aim to influence the main CPython implementation, the community around PyPy and Python was the main target for dissemination of the results. With this taken into account and considering the fact that the core practice for project coordination was through sprints, the newsletters for external stakeholders consisted of the sprint reports produced.

The sprint reports summarize the technical achievements of the sprints, who contributed to them and where the sprints were organized. During the project, 16 sprint reports have been produced.

The sprint reports have been distributed via email to pypy-dev (and Python mailinglists) and uploaded to the website, from where a list of all sprints and their reports is accessible.

5.3   Mailing lists and IRC channels

The following mailing lists have been used in the project:

- pypy-funding (Consortium members mainly)

- pypy-eu-svn (automated mailinglist tracking changes to files in the Consortium repository)

- pypy-manage (a recent mailinglist made for the Management Team to prepare issues that should be brought to the Consortium level for decision making)

- pypy-dev (the main development mailinglist) (324 users as of 2007-02-19)

- pypy-dev-svn (automated mailinglist tracking changes to files in the source code repository) (56 users as of 2007-02-19)

- pypy-sprint (mailinglist for coordinating sprint organisation and logistics)

The main mailing lists have accessible archives dating back to 2003. Backdating makes it easier for new contributors to enter the project and understand the background and evolution of the project.

The following IRC channels have been used in the project all hosted at freenode.net (URL: http://freenode.net):

---

[3]Find a more detailed description of the on-line documentation in report 14.2 *Tutorials and Guide through the PyPy Source Code* (14.2)

- #pypy-funding (channel for discussing Consortium issues)

- #pypy-manage (recent channel for preparing Consortium issues by the Management Team)

- #pypy-Consortium (recent non-public channel, used for Consortium meetings only)

- #pypy-sync (channel used only for pypy-sync meetings)

- #pypy-tb (channel for coordination and meetings of the Technical Board)

The main IRC channels have IRC logs/archives dating back to early 2005. Such logs are used for supporting and validating minutes of the meetings (pypy-sync, Technical Board and Consortium, a quality measure that was not part of the original Quality Assurance plan (1.1)).

6 References

7 Glossary of Abbreviations

The following abbreviations may be used within this document:

7.1 Technical Abbreviations:

AST	Abstract Syntax Tree
CPython	The standard Python interpreter written in C. Generally known as "Python". Available from www.python.org.
codespeak	The name of the machine where the PyPy project is hosted.
CCLP	Concurrent Constraint Logic Programming.
CPS	Continuation-Passing Style.
CSP	Constraint Satisfaction Problem.
CLI	Common Language Infrastructure.
CLR	Common Language Runtime.
docutils	The Python documentation utilities.
F/OSS	Free and Open Source Software
GC	Garbage collector.
GenC backend	The backend for the PyPy translation toolsuite that generates C code.
GenLLVM backend	The backend for the PyPy translation toolsuite that generates LLVM code.
GenCLI backend	The backend for the PyPy translation toolsuite that generates CLI code.
Graphviz	Graph visualisation software from AT&T.
IL	Intermediate Language: the native assembler-level language of the CLI virtual machine.
Jython	A version of Python written in Java.

LLVM	Low Level Virtual Machine - a compiler infrastructure available from University of Illinois at Urbana-Champaign
LOC	Lines of code.
Object Space	A library providing objects and operations between them, available to the bytecode interpreter via a well-defined API.
Pygame	A Python extension library that wraps the Simple Direct-Media Layer - a cross-platform multimedia library designed to provide fast access to the graphics framebuffer and audio device.
pypy-c	The PyPy Standard Interpreter, translated to C and then compiled to a binary executable program
ReST	reStructuredText, the plaintext markup system used by docutils.
RPython	Restricted Python; a less dynamic subset of Python in which PyPy is written.
Standard Interpreter	The subsystem of PyPy which implements the Python language. It is divided in two components: the bytecode interpreter, and the standard object space.
Standard Object Space	An object space which implements creation, access and modification of regular Python application level objects.
VM	Virtual Machine.

7.2 Partner Acronyms:

DFKI	Deutsches Forschungszentrum für künstliche Intelligenz
HHU	Heinrich Heine Universität Düsseldorf
Strakt	AB Strakt
Logilab	Logilab
CM	Change Maker
mer	merlinux GmbH
tis	Tismerysoft GmbH
Impara	Impara GmbH

References

(AGILE) Beatrice Düring, *Trouble in Paradise: the Agile Open Source Project PyPy, EU-funding and Agile Practices*, AGILE 2006, Minneapolis, USA

(14.2) *Tutorials and Guide Through the PyPy Source Code*, PyPy EU-Report, 2007

(14.5) *Documentation of the Development Process*, PyPy EU-Report, 2007

(2.1) *Development Tools and Website*, PyPy EU-Report, 2007

(1.1) *Create QA Plan for the Project*, PyPy EU-Report, 2005

(PER) *Periodic Activity Report* (Reporting Period II), PyPy EU-Report, 2007

(FIN) *Final Activity Report*, PyPy EU-Report, 2007