

Business Case of the Project: APSP

Yaroslav Halchenko^a, Christopher Rebstock^b, Kareemullah Quadir^c,

Roshan Rammohan^d, Vinod Doshi^e

February 16, 2002

Overview

The Personal Software Process is a software development process that allows the individual to apply industrial level discipline to his or her practice.

Our objective is to develop a APSP (Automated Personal Software Process) tool, which would be used together with CVS to automate the PSP process to a large extent.

Purpose

To assess personal achievements and progress it is useful to have a tool which would show trends of stages where a developer spends most of their time.

Most of PSP tools existing on the market ([1] has list of links to most of them) involve filling long forms. This process can be time consuming and could demotivate people from using PSP and may not report some aspects of Software development.

Vision

The idea is to create a tool in which the user can get the most out of the personal software process with the least effort. We will accomplish this by gathering as much data as possible automatically. CVS wrappers and background tools will gather most data without user intervention.

Special Features

- Automatic and constant file monitoring.
- Client/Server Architecture would enable a developer 'roaming access' to his PSP.
- The 'Thin Client' would be platform independent.

Server

Duties of the server:

- collect information from all users
- provide web-backend analysis of data such as
 - Projects he was working on.
 - How much time a programmer spends on specific sub-tasks in a project.
 - Statistics on specific files he was working on.
 - Data on CVS transactions(commit, update, checkout)

Client

The Client will collect information and send it to the Central APSP server.

Kinds of clients:

- Monitoring-tool which would run all the time to monitor user's activity on files in specific directory.
- set of CVS wrappers to report activity on the project in CVS
- may be some other tools to grab other kinds of statistics (register number of compilations and debugging time etc).
- small tool which would report change of developers activity (developing, debugging etc)

OS and Hardware Requirements

The system will be developed and tested on the Debian 2.4 Linux distribution and hardware capable of running this would suffice.

Tool depends on the presence of CVS client on current machine (and of course CVS server anywhere - no direct transactions with CVS server will be made) and assume that projects which developer works on kept within that CVS so monitoring program can get that directory name and get list of files and subdirectories from CVS/Entries.

Development and delivery platforms

Database: MySQL

Web Server: Apache

Development tools: Shell Programming, Java(Client Server),
Javascript(web client-side scripting), HTML

Risks

1. Significant portion of the group is not well versed in shell programming, databases and CVS.
2. We're still finding out what PSP is all about.
3. The idea of the authentication procedure is hazy.
4. Integrating various technologies/platforms/software that we plan to develop this tool might prove tricky.
5. Analysis of file modification under development (how many lines were added/deleted/edited) can be a tricky algorithm.
6. Defining a protocol for communication between clients and server can be hard (needs to provide caching mechanism for cases when client is off network).
7. Might not complete the project in time.

References

- [1] Psp resources page at the university of karlsruhe. URL <http://www.ipd.ira.uka.de/PSP/>.