Joshua Tasman
February 1, 2000
CS190
Requirements Document, ver. 1

Large Project Managment System

goal

Creatation of a development tool specifically for working with and managing large, multiuser collaborative projects.

overview/introduction

The benefits of working on a large project with a group of fellow developers is well known. Thanks to the phenomenon of the internet, many large collabortive projects have been sucessful. In many such projects, it is common that most of the team members have never even physically met. These types of projects have included well known examples such as the language PERL and the operating system LINUX. Many similar "open-source" projects of late have sprung up. The system of RCS/CVS has been established as a mechanism for library management. However, as far as *social* aspects of manganaging the project are concerned, there is much less of a concensus. Even with a moderately high-profile project such a major company's attempt at working on a driver for a sound card (http://opensource.creative.com/pipermail/emu10k1-devel/2000-January/thread.html), no specfic management tools are used besides RCS/CVS. There seems to be an unfilled niche for an *unobtrusive* development tool which not only integrates awareness of the unique challenges of multiuser development, but is designed from the group up with this awareness in mind.

user experience and requirments

required features

- The system will maintain a database of people on the project, with information such as contact information.
- Collaborative timetables can be input into the system. These can be accessed by anyone on the project. The users can
  - take surveys, and rate their confidence in upcomming milestones;
  - add comments to the calendar.

  Users can sign up to receive reminders or updates by email.

- The system will provide a graphical front-end to the RCS/CVS file library system, along the lines of the preexisting system “LinCVS” (http://ppprs1.phy.tu-dresden.de/ trogisch/lincvs/lincvsen.html).

- The user is able to see a list of what files for her or his project are currently active. People can report on their state (with both text comments and a rating scale), and others can visually observe what stage each person and/or the entire project is at. The information could, for example, be provided by a visual cue with the use of color to indicate project status at a glance.

- The interface to the system must be user-configurable, to display any combination of information in an intuitive and concise manner. The interface must not take up so much screen space that a developer could not keep it as the “top” window on the display while using her or his usual development tools. (see next item)

- as an alternative to a graphical display, a terminal-based command will provide a small summary of previously selected data.

- Privacy issues must assured: the user must be able to control access to his or her current state and personal data (such as contact information).

- A utility should be provided so that non-private data can be converted to HTML format for publishing on the web.

optional features

- In addition to the HTML conversion utility, a web-based interface could be provided.

- In general, we want to avoid re-inventing the wheel. For such utilities as instant messaging and news mailers and readers, we should encourage use of existing systems. An optional consideration for extensibility would be to design the system so that the user could pick an editor, debugger, news reader, &c., and our system would provide a tight integration between the components for a total IDE.

- An optional utility could be to provide a tool to do simple “physical” design checking (as described in Lakos).