Component Assessment Server (CAS)
PROJECT PLAN

27th October 2004

Project period: September 15th 2004 - March 15th 2005
Student: Samuele Milani (milanissa@student.ethz.ch)
Student-No: 99-918-880
Supervising Assistant: Till G. Bay
Supervising Professor: Prof. Bertrand Meyer
1 Project description

1.1 Overview

The quality of software components is very important, because components are used to build bigger software systems. Many different factors contribute to component quality for example the components acceptance, its behaviour, the constraints it satisfies, how it is designed and how extendible it is (see "The Grand Challenge" [8] for more detailed discussion of these quality attributes). Techniques like Design by Contract [1] and the Object-Oriented programming paradigm help to improve quality.

No Software project is perfect from the start, but the quality attributes of it evolve. Therefore, assessment of quality attributes needs to be done regularly in order to allow observing the evolution of these attributes.

The goal of this project is to build a system that allows automatic assessment of quality attributes. We talk about the scope of the system in the next section.

1.2 Scope of the work

The project is divided into three principal subprojects: the Component Assessment Server, the Component Assessment Server configuration module and the extensions for the Component Assessment Server.

Component Assessment Server

The server should be able, starting from a given configuration, to connect to a source code repository and get the latest version of one or more software projects. Additionally it should be able to launch one or more extensions to measure different quality attributes of these projects and send the results of such an assessment to the users.

Component Assessment Server configuration module

All the configuration of the Component Assessment Server is done with a Web Interface. The purpose of this module is to allow creating new project configurations for the server and to allow scheduling and starting extension runs.

Extensions

The Component Assessment Server should allow measuring different quality attributes using different extensions.
1.3 Intended results

We start with an extension that will work as a continuous integration server that reports success or failure of test cases. This extension will be one implementation of a quality attribute that can be measured and another goal of this project is to provide an extension interface that allows plugging in other component assessing extensions. To test this extension interface we will develop an extension that calculates various metrics.

1.3 Intended results

The intended results of the master thesis are:

- An extendible component assessment server
  A server that allows automatic assessing of components, the server can be used to assess many different quality attributes of a component. It should be easy to add new extensions to the server to assess other quality attributes.

- Implement various extensions
  Implement an extension that works as a continuous integration server, and a test extension to calculate various metrics like, lines of code, number of interfaces...

2 Background material

Reading list

- OOSC2 [1], in particular following chapters:
  - Chapter 1: Software quality
  - Chapter 11: Design by Contract: Building reliable software

- Continuous Integration [2]
- Gobo documentation [3]
- CVS documentation [4]
- e-Posix documentation [7]

Tools

- Eiffel Studio 5.5
- Gobo
- e-Posix
- PHP
3 Project management

3.1 Objectives and priorities

<table>
<thead>
<tr>
<th>Objective</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design of the system</td>
<td>1</td>
</tr>
<tr>
<td>Read configuration</td>
<td>2</td>
</tr>
<tr>
<td>Communication with repository</td>
<td>1</td>
</tr>
<tr>
<td>Continuous integration extension</td>
<td>2</td>
</tr>
<tr>
<td>Reporting results</td>
<td>2</td>
</tr>
<tr>
<td>Service implementation</td>
<td>1</td>
</tr>
<tr>
<td>Schedule implementation</td>
<td>1</td>
</tr>
<tr>
<td>Configuration module</td>
<td>3</td>
</tr>
<tr>
<td>Extension</td>
<td>2</td>
</tr>
<tr>
<td>Other extensions</td>
<td>3</td>
</tr>
<tr>
<td>Thesis report</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1: Objectives and their priorities (1=high 2=middle 3=low)

3.2 Criteria for success

While the ultimate goal of the project is the creation of a complete component assessment server, the criteria for success is the quality of the software and the documentation handed in at the end. The result may be a partial implementation of the above objectives without implying any penalty on the success of the project. In this respect, the quality of the software is measured according to the following points:

- use of Design by Contract
  - pre- and postconditions
  - class invariants
  - loop variants and invariants

- careful design
  - design patterns
  - extendibility
  - reuse-ability
  - careful abstraction

- core principles of OOSC2
  - command/query separation
  - simple interfaces
  - uniform access
  - information hiding
3.3 Method of work

- style guidelines (from OOS2 [1], and from Gobo [3])
- correct and robust code
  - test extension working
- readability of the source code

And the quality of the documentation is measured according to the following points:

- completeness
- understandability
- usefulness
- accessibility and structure

3.3 Method of work

We have chosen to implement our CAS server with ISE EiffelStudio 5.5 and the configuration module with PHP.
The Component Assessment Server should be able to work with a CVS server [4], but we plan to make it communicate with other source code repositories too (e.g. Subversion). For building the sources we use gnat and for testing the build getest (included in Gobo [3]).

3.4 Quality management

The quality will be ensured by (in descending order of importance):

- Close contact with the supervisor, recurring discussions about the design of CAS
- The creation of relevant test-suites
- The documentation

3.4.1 Documentation

The documentation will be written throughout the whole project. Each of the project steps will be reflected as a chapter in the thesis report. The Component Assessment Server itself will be documented on one hand in the source code and on the other hand in the Component Assessment Server manual. The whole documentation will be comprised in the thesis report.

The thesis report will be exposed to constant review by the supervisor.
3.4.2 Validation steps

The validation steps are guidelines for the recurrent events that allow me to validate my progress along the project plan. If it is not possible to have a meeting with my supervisor on a weekly basis, there should be a meeting at least at the end of each project step (see below). The ending dates of the project steps form at the same time the partition of the project into milestones.

- Weekly status report to supervising assistant, either via e-mail or personally
- Intermediate presentations of the work during group meetings
- Final presentation

4 Plan with milestones

4.1 Project steps

- **Startup**
  - Feasibility study
  - Decision to extend an existing project or to create a new project
  - Choice of technology
  - Basic design of the system
  - Finish project plan
  - *Milestone M1: Project Plan. Friday, October 8th, 2004*

- **Communication with the source code repository**
  - Reading the configuration file
  - Communication with source code repository
  - Performing a ‘checkpoint’ of a project
  - *Milestone M2: Correct checkout, Friday, November 12th, 2004*

- **Component Server Assessment implementation**
  - Creating the server
  - Implementing the schedule
  - *Milestone M3: Working Component Assessment Server, Friday, December 17th, 2004*

- **Reporting of extension output**
  - Automatic compilation, linking and testing of the software project.
  - Continuous integration extension
  - Reporting of the test results
  - *Milestone M4: Results reporting and continuous integration extension, Friday, January 14th, 2005*

- **Configuration interface**
  - Implement a web-based interface with PHP
- Configuration interface creating server configurations
- Configuration interface showing report results
- **Milestone M5: Configuration interface, Friday, January 28th, 2005**

**Extensions**
- Metrics calculation
- Other extensions
- **Milestone M6: Metrics calculation extension, February 18th, 2005**

**Documentation**
- Thesis report
- Server user guide
- Client user guide
- **Milestone M7: Documentation and user guides, March 15th, 2005**

### 4.2 Deadline

**Project start:** Wednesday, September 15th, 2004

**Project end:** Wednesday, March 15th, 2005

### 4.3 Tentative schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Startup</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Procedure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technology</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Milestone M1: Project Plan</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Communication with proxy server</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proxying the configuration file</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication with source code repository</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performing a validate of proxy and Milestone M1: Control plan</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Component server Assessment implementation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creating the server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Milestone M2: Component Assessment Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Reporting of extension output</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compilation, testing, and testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reporting of the test results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intermediate WIP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Configuration interface</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interface and web server interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration interface creating server configurations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration interface showing report results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Milestone M3: Configuration interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Extensions</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extensions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other extensions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Milestone M4: Extensions</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Documentation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thesis report</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server user guide</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client user guide</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Milestone M5: Documentation and user guides</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Presentation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final presentation of project</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Tentative project schedule
References


    Available at http://www.martinfowler.com/articles/continuousIntegration.html

    Available from http://www.gobosoft.com

[4] Per Cederqvist: *Version Management with CVS.*
    Free Software Foundation, Inc. 2004

    Available at http://lerdorf.com/veracruz.pdf

[6] Peter Flynn: *The XML FAQ.*
    Available at http://www.ucc.ie/xml/faq.html

    Available at http://www.berenddeboer.net/eposix/

    Available at http://csdlcomputer.org/comp/proceedings/icse/2003/1877/00/18770660abs.htm