

CENG 492

D&D SOFTWARE

**CONFIGURATION
MANAGEMENT
PLAN**

Prepared by:
Firat Alpergin
Dogan Yazar
Tuncay Namli
Mehmet R. Dogar

TABLE OF CONTENTS

1. INTRODUCTION	3
1.1 Purpose.....	3
1.2 Document Overview	3
1.3 Scope.....	4
1.4 Benefits	4
1.5 Reference Documents	4
1.6 Acronyms and Abbreviations	4
2. ORGANIZATIONAL FRAMEWORK.....	5
2.1 Roles & Responsibilities.....	5
2.2 Resources	6
2.3 Policies, Directives and Procedures.....	7
3. CONFIGURATION MANAGEMENT TASKS	7
3.1 Configuration Identification.....	7
3.2 Configuration Items	8
3.3 Baseline Management.....	9
4. CONFIGURATION CONTROL.....	10
4.1 Change Control	10
4.2 Version Control.....	11
4.3 Release Management	11
5. CONFIGURATION AUDITS AND REVIEWS	11
5.1 Functional Configuration Audits	11
5.2 Peer Reviews.....	12
5.3 Tools	12

1. INTRODUCTION

1.1 Purpose

Software development is an activity that requires the management of changes. Actually, the changes and development you make has no value if you do not have a plan to manage them. This becomes clearer when the size of the software gets bigger, and the numbers of developers get higher. Change management determines your end product's quality and whether it meets the specified deadlines or not. In this aspect, software configuration management plan becomes one of the crucial tasks in the software development lifecycle.

This document is the Software Configuration Management Plan of D&D Software Company for the project management tool DProject. Here, we try to establish an integrated process for identifying, documenting, monitoring, evaluating, controlling, and approving all changes to work products throughout the lifecycle of our project management software.

1.2 Document Overview

This document comprises of the following sections:

- *Introduction:* This section explains the purpose and scope of this document, identifies the benefits that it will provide, and establishes the relationship of this document with other documents;
- *Organizational Framework:* This section explains the responsibilities related to the CM tasks for the project and gives information about the individuals responsible for these tasks;
- *Configuration Management Tasks:* This section identifies the configuration management tasks and items, establishes the baseline management procedure, identifies the main baselines, and describes the repositories that are used;
- *Configuration Control:* This section describes the evaluation procedure of all change requests and change proposals and their subsequent approval or disapproval;
- *Configuration Audits and Reviews:* This section describes the audit and review procedure following the changes, milestones, and baselines; so that every member of the development team will be aware of these changes, can comment on them and can contribute to their approval or disapproval.

1.3 Scope

The scope of this document encompasses all configuration management activities for DProject. It identifies responsibilities of all members regarding configuration management, describes all configuration management tasks, describes all policies and procedures for making a change and the audits and reviews following them.

Every D&D Software member has to pursue the conventions and procedures defined here when naming a file, making a change, and saving these changes to the repository.

1.4 Benefits

This Software Configuration Management Plan will contribute to the development of DProject in the following ways:

- The integrity of the system will be maintained by the dependency tracking of the changes made on the configuration items.
- Coordination of access to configuration items will be managed, so that concurrent changes will not conflict,
- Changes will be controlled and managed effectively, so that only necessary and sufficient changes will be realized.
- Communication throughout the developers will be improved by reporting and auditing.
- As the consequence of all these, overall project quality will be improved by effective configuration management.

1.5 Reference Documents

The documents referenced in this report are:

- “Software Requirement Specification” for DProject
- “Detailed Design Report” for DProject

They can be found at our website <http://senior.ceng.metu.edu.tr/2005/ddsoftware> .

1.6 Acronyms and Abbreviations

CM: Configuration Management
CI: Configuration Item
RA: Requirements Analysis
FBL: Functional Baseline

ABL: Allocated Baseline
PBL: Product Baseline
FCA: Functional Configuration Audits
CVS: Concurrent Versions System

2. ORGANIZATIONAL FRAMEWORK

2.1 Roles & Responsibilities

This section explains the responsibilities related to the CM tasks for the project and gives information about the individuals responsible for these tasks.

There are three organizational CM units:

- Configuration Management Team (CMT)
- Configuration Control Board (CCB)
- Engineering Review Board (ERB)

2.1.1 Configuration Management Team (CMT)

CMT is responsible for the CM related activities of the project. It can be thought as the administrative unit for CM activities. CMT of our project is composed of Mehmet Remzi Dogar and Firat Alpergin. The basic responsibilities of CMT are as follows:

- Creating and maintaining the CM plan of the project.
- Checking (and forcing, if necessary) for the effective implementation of the activities stated in the CM plan.
- Coordinating the basic CM activities throughout the project.
- Making the final decision on the subjects (which are discussed by the whole team) related to the CM activities of the project.

2.1.2 Configuration Control Board (CCB)

CCB is the unit that either accepts or rejects the change control requests. These requests include subjects like design specifications, schedules, hardware usage, etc. The requests related to the CM plan of the project are not decided only by the CCB, but also with CMT. Also, CMT and CCB interact with each other in case of some changes that may impose an unavoidable CM change. CCB of our project is composed of Tuncay Namli and Dogan Yazar. The basic responsibilities of CCB are as follows:

- Reviewing and either approving or rejecting the changes related to the project (except the one that are explained above).
- Analyzing (and trying to reduce, if necessary) the impacts of the changes made (in cooperation with ERB).
- Setting and maintaining a standard procedure for keeping track of the changes related to the project.

2.1.3 Engineering Review Board (ERB)

ERB is responsible for providing the technical analysis of the possible changes to the system. It is naturally inclined to CCB and therefore works closely with it. ERB of our project is composed of Tuncay Namli and Dogan Yazar. The basic responsibilities of ERB are as follows:

- Analyzing the possible complexities (with an engineering point of view) induced by a possible change in the project.
- Setting and maintaining a standard procedure for analyzing and explaining the results of possible changes.
- Analyzing (and trying to reduce, if necessary) the impacts of the changes made (in cooperation with ERB).

2.2 Resources

The resources needed for performing the CM tasks include personnel, training, hardware, and software (also for documentation). We all have these resources as explained below:

- The personnel for the CM tasks are the same 4 members of our project since this is not a large-scaled project and we don't need extra staff.
- The training and knowledge is obtained by all members of our staff through Ceng350 - Software Engineering course.
- The necessary hardware is provided by our department.
- The necessary software is also provided by our department. The software used for documentation is Microsoft Office. Also, CVS will be mainly used for the CM tasks. CVS will keep track of the changes made in the project history and will be useful in version management. Check-in and check-out facilities will make it possible to return to a previous point in project history undoing changes. Making all the members use the same tool will enable an efficient interaction among them.

2.3 Policies, Directives and Procedures

There are some predefined rules that are to be applied throughout the project and these rules are strictly followed when a user wants to make a change related to any aspect of the project. These procedures are as follows.

When a user wants to make a change in the project (apart from a change in the CM plan), he first applies to the CCB. Then CCB reviews the change suggested by the user and if necessary, ERB analyzes the impacts of the suggested change. Depending on the results of ERB, CCB makes a final review and makes a decision. Then the user is notified of the result.

If a user wants to recommend a change in the CM plan, he applies to CMT. CMT reviews the suggestion independently and makes a decision. Then the user is notified of the result.

3. CONFIGURATION MANAGEMENT TASKS

Since DProject is a web based project that uses three-tier model, we must have configuration management tasks that will include all these tiers and the dependencies between them. These tasks mainly includes the control of the configuration, identifying and resolving dependencies, notification of all members and documentation of the configurations to ensure the coordination between group members.

3.1 Configuration Identification

This section lists the categorization of the configuration items to examine them systematically.

- **Communication Items**

Communication is very important to notify the group members about the configurations done for the project. Communication items will be used for all type of configuration on the project. All of the configuration documents will be shared among the team members using the mail group that we started. Each team member is responsible for posting all his configuration documents to the group.

- **Documentation Items**

Documentation is used to archive the configuration details to serve up to the group members and to support the configuration management process.

- Hardware Items

Hardware items are used for archiving the configuration documentation and the project documents.

- Software Items

The most important issue in the project is the management of configurations on software. We can divide the software items as follows:

- Database
- Application code
- Software for Presentation

3.2 Configuration Items

This section shows our agreed configuration items and their identification. Every subsection defines description, usage, identification and processes for the items when configured. Versioning of the documents is discussed at the end of this section.

3.2.1 Database

Database is the crucial component of our project since all processes starting from the login to the end need database. Another reason which increases the importance of database for configuration management is that the changes in structure of the database will easily affect the whole structure. There are two aspects in this issue as structure management and the test data. We use a file that includes sql operations to construct the database for the project. Configuration for the database structure must be done on this file. However, configuration for this file must be discussed and agreed by the all members because of the possible inconsistencies. After the configuration, the configuration must be documented and send to the documentation archive. File name for this file is 'createdatabase.sql' and should not be changed to distinguish it form the other .sql files. Test files can be different for all members for their special tests but for constructing sql files for the overall testing procedure, these must be documented and combined by one of the members for some time intervals. These files must include the goal of the file and the creator as a comment in the file.

3.2.2 Data Files

Data files that will be used in DProject are the image files for presentation items, resource bundle that includes the static text in the presentation layer. These files can be configured without discussion but they must be versioned to decide the final structure of the presentation layer. Images are either .gif or .jpeg files and must have content related names and every word in their names must be separated by “_”.

3.2.3 Software Configuration

Software configuration management is the other crucial concept for our project. Although CVS is used to share the software consistently, we have some tasks for this issue. There are three different software groups in our projects. One of them is the data beans and application classes which must be more robust to the configurations since all presentation and middleware use these components. Configurations to these classes must be documented and all other members must be notified from the configuration to enhance the identification of inconsistencies. Members must be careful to obey the general interface of the class and must check the possible inconsistencies and add these to the document of configuration in case of interface changes. Middleware includes the servlets that handle the preprocessing and routing between presentation pages. Servlets must be named with the name of the corresponding jsp page name with adding ‘*Servlet*’ at the end of the name. Since servlets can be used to serve more than one page, configuration must be done by considering these dependencies and the member responsible for the component must be notified for the configuration. Presentation layer includes the jsp pages that show the data provided from middleware and other components. These pages must be named by considering the content of the page and words must be separated by “_”. Configuration on these pages must be documented and if the presentation structure of the page is changed the .htm form of the page is documented and sent to the members to notify them.

3.3 Baseline Management

Baselines are composed of CIs at a specific point in time. The baselines are used to control changes to the CIs throughout the life cycle of DProject. DDSOFT authorizes the creation of new baselines and approves any changes to a baseline. Baselines will be tracked, audited, retained and version controlled in the Concurrent Versions System (CVS) version control tool.

There are four different types of baselines for DProject: functional, allocated, product. Product baseline includes all of the software components required to run the DProject system in a production environment.

The baselines and their contents are as follows:

- **Functional Baseline (FBL):** A functional baseline is established following the Requirements Analysis (RA) and delivery of required documentation. Follow the web site of DDSOFT for RA as the Requirement Analysis Report. Every member of DDSOFT is responsible of verifying all required documents and configuration items, related to her developments, are established properly.

- **Allocated Baseline (ABL):** An allocated baseline is established at the end of the design phase (prior to development). It is verified that all required documents and configuration items are established properly. The CIs in the allocated baseline include all preliminary system designs, detailed system designs, and associated test plans. The documentation about ABL is located on the web site of DDSofT as Initial Design Report and Detailed Design Report.
- **Product Baseline (PBL):** A product baseline is established for each release of the system at the end of the acceptance testing phase. The developer(s) verifies that the tested product is exactly described in the product baseline documents. It includes all CIs, CM data, system test plan reports, manuals, and other plan documents. For the documentation about PBL, the web site of DDSofT can be followed.

4. CONFIGURATION CONTROL

4.1 Change Control

In every stage of the development of the project, there may be a need to make some changes on it. But after these changes are done, some conflicts may appear between the newly added code and the previous codes. In order to avoid these conflicts, a procedure must be defined among the team members and all of the change requests must be handled according to this procedure. Since our projects are not large-scaled and our team is relatively small (only 4 people), change control, which includes the evaluation of all change requests and their approval and their following approval and disapproval, is easier for us. But still it's very important and we defined a process for this task which is:

All the changes, no matter how insignificant are they; will be reported and the requests and the results of them will be documented.

For the minor changes that do not affect the interfaces of our classes;

- They will be fixed by the team member who implemented it,
- No other team member has the right to do the minor change if he didn't implement the code.

For the major changes;

- Each team member must document his request about what he wants to change including the reason, priority and the definition of the change.
- A team meeting will be held and to discuss about the change request and approve or disapprove this request.

- If the change is found necessary, it will be approved and the change will be implemented by the relevant person and team members will verify the correctness of it.
- If the change is found unnecessary, the request is rejected and all group members are informed about it.

4.2 Version Control

We will use the CVS server provided by our department as our version control tool which is a “Revision Control” tool designed to keep track of the changes on the source files by groups of developers. All of our configuration items will be kept in repository and we will have local copies that we will back up them frequently. CVS automatically does a version control on source files for us. Here are the steps of how we will manage version control via CVS:

- 1) A member checks out a file.
- 2) He changes his copy of the file.
- 3) After he is done with the changes, he uses CVS commit to merge his changes into the repository so these modifications will be available to all users.
- 4) CVS increments the version of the file automatically.

4.3 Release Management

Release management, which is the process through that software is made available to its users, is an important task for any maintenance project. Our project will have two releases, one initial and one final. Depending on the feedback we receive from our instructors and assistant after the first release, we will finalize the features for our final release.

5. CONFIGURATION AUDITS AND REVIEWS

5.1 Functional Configuration Audits

The Functional Configuration Audits (FCAs) determines whether the actual implemented performance of each configuration item complies with its design specifications. It verifies that the proposed functional design baselines are consistent with the changes.

The FCAs will be held as a part of D&D software weekly meetings. Every major change request will be discussed in an FCA; and will be realized only if it is seen as

consistent with design specifications and baselines. This will protect the software team from doing changes that does not comply with the specifications.

5.2 Peer Reviews

Peer reviews are the procedures by which the qualities of the changes are evaluated by other people outside of the developer team.

Our weekly meetings with our teaching assistant Levent Bayindir will take place as peer reviews, where he will evaluate our changes and comment on them. These reviews will supply us with point-of-views that come outside of our team; and this way we will manage to make change decisions on a more objective basis.

5.3 Tools

The tools that will facilitate our audit and review processes include the CVS, and our mail group.

CVS will act as the major sources of the configuration items, which will be under discussion. Every audit will also rely on CVS as the repository of past versions.

D&D software mail group will also be used to exchange comments, and decisions on changes.