Configuration Management Plan
for
Watch & Touch

by DialecTech
Giray Havur (1630870)
Melike Ercan (1560135)
Utku Şirin (1560838)
Yaman Umuroğlu (1560614)
# Table of Contents

1. Introduction.......................................................................................................................... 3
   1.1. Purpose of Configuration Management Plan................................................................. 3
   1.2. Scope of the Document................................................................................................. 3
   1.4. Document References................................................................................................. 4
   1.5. Document Overview..................................................................................................... 5
2. The Organizations CM Framework....................................................................................... 6
   2.1. Organization................................................................................................................ 6
   2.2. Responsibilities........................................................................................................... 6
   2.3. Tools and Infrastructure............................................................................................... 7
      2.3.1. Software Development........................................................................................... 7
      2.3.2. Version Control...................................................................................................... 7
      2.3.3. Project Management.............................................................................................. 7
      2.3.4. Libraries................................................................................................................ 7
3. Configuration Management Process.................................................................................... 8
   3.1. Identification................................................................................................................ 8
      3.1.1. Hardware Components.......................................................................................... 8
      3.1.2. Source Code.......................................................................................................... 9
      3.1.3. Documentation...................................................................................................... 9
   3.2. Configuration Management and Control....................................................................... 10
      3.2.1. Software System Change Requests........................................................................ 10
      3.2.2. System Change Evaluation.................................................................................... 11
      3.2.3. System Change Implementation............................................................................. 11
   3.3. Configuration Status Accounting................................................................................ 11
   3.4. Auditing........................................................................................................................ 12
4. Project Schedules and CM Milestones................................................................................ 12
   4.1 Project Schedules........................................................................................................... 12
   4.2 CM Milestones.............................................................................................................. 12
5. Project Resources................................................................................................................ 14
   5.1 Budget............................................................................................................................ 14
   5.2 Manpower..................................................................................................................... 15
   5.3 Software Tools............................................................................................................. 15
6. Plan Optimization................................................................................................................ 15
1. Introduction

This document is the Configuration Management Plan of Watch & Touch project, created by the DialecTech team. In short, the aim of Watch & Touch is to create a complete, free & open source, multi-platform, interactive whiteboard system which supports multi-touch gestures, complemented by an in-classroom collaborative working client to increase student participation in classroom activities.

1.1. Purpose of Configuration Management Plan

The purpose of this document is to define and explain the CMP for Watch & Touch, which consists of two main components, named the IWBC and the CBC. Each component in the project has numerous different modules that cooperate or are related to each other, which requires careful planning and execution during the development process - from the very start of the project, DialecTech, has been aware of the importance of carefully following the steps of software development process and has given a high degree of importance to both quality in documentation and in implementation. As a crucial part of this process, we put forward our efforts on Configuration Management, that is the development and use of standards and procedures for managing an evolving software system[2]. CM procedures define how to record and process proposed system changes, how to relate these to system components and the methods used to identify different versions of the system. CM is considered as a part of software quality management and the definition and use of CM standards is essential for quality certifications in both the ISO 9000 and the CMM and CMMI standards that are worldwide valid and necessary for software products. CM tools are used to store versions of system components, build systems from these components and track the releases of system versions to customers. Hence, the Configuration Management Plan is an essential part of documentation for overall improvement of our software system, as it helps us to prevent conflicts between team members and help towards the application of possible changes during development, as well as being useful for recognizing relationships between system components in a structural way from a development perspective.

To sum up, CMP allows software developers and software project managers to control and identify causes and results of changes in the software development process in order to implement these changes. Guidance of CMP is very important for a software engineering team’s communication and coordination efficiently.

1.2. Scope of the Document

The scope of this document is the determination of a high level CMP for Watch & Touch project of DialecTech. CMP presents configuration management activities that will be applied throughout the development process of our project and these activities explained in this document are applicable during the development and maintenance phases of the project. Basically, this document consists of methodologies and activities designed to manage changes and relate them. Coping with versioning mechanism, standardization, milestones, and an overview of utilized resources for the Watch & Touch project are included, as well. The organization of our team (DialecTech), and the schedule of the project are the other
issues that will be mentioned in the document. The related audience for this document is DialecTech software team, and the Department of Computer Engineering faculty as part of the Senior Computer Engineering Design (CENG492) course.

1.3. Definitions, Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWBC</td>
<td>Interactive White Board Client</td>
</tr>
<tr>
<td>CBC</td>
<td>Collaboration Client</td>
</tr>
<tr>
<td>CM</td>
<td>Configuration Management</td>
</tr>
<tr>
<td>CMP</td>
<td>Configuration Management Plan</td>
</tr>
<tr>
<td>SDT</td>
<td>Software Development Team</td>
</tr>
<tr>
<td>STT</td>
<td>Software Testing Team</td>
</tr>
<tr>
<td>CMT</td>
<td>Configuration Management Team</td>
</tr>
<tr>
<td>RCT</td>
<td>Release Control Team</td>
</tr>
<tr>
<td>CMM</td>
<td>Capability Maturity Model</td>
</tr>
<tr>
<td>CMMI</td>
<td>Capability Maturity Model Integration</td>
</tr>
<tr>
<td>CR</td>
<td>Change Requests</td>
</tr>
<tr>
<td>SCR</td>
<td>System Change Request</td>
</tr>
<tr>
<td>CVS</td>
<td>Concurrent Versions System</td>
</tr>
<tr>
<td>SVN</td>
<td>Subversion</td>
</tr>
</tbody>
</table>

1.4. Document References

In chapter 29 (Configuration Management) of this source, a lot of vital information about CM and CMP is given.

A common software engineering standard to provide some guidance and recommended approaches for specifying software design descriptions

The SDD for Watch & Touch, prepared according to the IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications

Homepage of the Watch & Touch project on Google Code
Qt Meta-Object Language is a JavaScript-based, declarative language for designing user interface centric applications.

Watch & Touch project page and blog

Qt Creator cross-platform C++ integrated development environment which is part of the Qt SDK

Subversion (SVN) is a version control system initiated in 2000 by CollabNet Inc.

Qt is a cross-platform application framework widely used for GUI applications.

The Poppler Qt4 interface library

The Wiiuse sourceforge.net

Google Code issue tracking system for Watch & Touch

1.5. Document Overview

This CMP document is divided into six sections, Introduction, Organization, Configuration Management Process, Project Schedule, Project Resources and Plan Optimization. The organization of this document is as follows:

- **Introduction**: This section consists of purpose and scope of the CM, some definitions, acronyms and abbreviations, document references.
- **Organization of CM Framework**: This section includes organization and responsibilities of DialecTech. In addition, it gives an overview of the tools and the infrastructure of the process.
- **Configuration Management Process**: This section describes the configuration management process to be used.
- **Project schedules and CM Milestones**: Important dates and milestones are listed in accordance with the CENG492 course deliverable.
- **Project Resources**: This section describes the resources utilized towards the completion of the project.
- **Plan Optimization**: This section explains methods which are going to be used to optimize the CMP.

5
2. The Organizations CM Framework

2.1. Organization

In work distribution, it is important for success to deliver the responsibilities to team members according to their previous experience and areas of interest. DialecTech consists of four people who are focused on different aspects of the project development according to their previous experience and areas of interest. The members with their main positions/titles in the team are as follows:

- Giray Havur - Graphical User Interface Designer, Developer
- Melike Ercan - Project Manager, Developer
- Utku Şirin - Software Architect, Developer
- Yaman Umuroğlu - Software Architect, Head Developer

Here are explanations of subcategories of DialecTech:

Development Team
Development Team is mainly responsible for implementing the modules of Watch & Touch project and making the changes that are requested by Testing Team. This Team will also be responsible from the releases.

Testing Team
Testing team will be responsible for two main purposes. First purpose is to check whether the modules meet the requirements. Second purpose is to give feedback about the modules and want to CR if necessary.

Configuration Control Team
Configuration Control Team will supervise all the activities of other groups. However, the main responsibilities of this group are review SCRs, accept or reject SCRs and monitor SCRs.

Configuration Management Team
Configuration Management Team is responsible for maintenance of the CM organization. Hence, this group will keep the CMP up-to-date.

2.2. Responsibilities

In our project, software system development process is shared by team members according to their responsibilities in the project.

Development Team: Yaman Umuroğlu, Utku Şirin, Giray Havur
Testing Team: Melike Ercan, Giray Havur
Configuration Control Team: Yaman Umuroğlu, Utku Şirin
Configuration Management Team: Melike Ercan
2.3. Tools and Infrastructure

2.3.1. Software Development:

Qt Creator: [7] Qt Creator is a cross-platform C++ integrated development environment which is part of the Qt SDK. It includes a visual debugger and an integrated GUI layout and forms designer. The editor's features includes syntax highlighting and autocompletion, but not tabs[3]. Qt Creator uses the C++ compiler from the GNU Compiler Collection on Linux and FreeBSD. On Windows it can use MinGW or MSVC with the default install and can also use cdb when compiled from source.

Qt SDK: [9] Qt is a cross-platform application framework that is widely used for developing application software with a graphical user interface (GUI) (in which cases Qt is referred to as a widget toolkit), and also used for developing non-GUI programs such as command-line tools and consoles for servers. It includes a wide variety of functionality ranging from container classes to GUI widgets which allow for rapid construction of feature-rich and stable applications.

2.3.2. Version Control

SVN: [8] Subversion is used to maintain current and historical versions of files such as source code, web pages, and documentation. Its goal is to be a mostly-compatible successor to the widely used CVS. SVN is a version control system that maintains current and historical versions of files such as source code, web pages, and documentation and stores these file in a server. We will be using the SVN repository provided by Google Code for its integration with the issue tracking system and to keep the source in a more public location.

2.3.3. Project Management

Google Code Issue Tracking System: [12] The issue tracking system present in Google Code is used for creating and maintaining tickets regarding development issues, as explained in detailed in section 3.3. We opted out on using Trac as we prefer the simpler interface and automatic integration to the Google Code SVN repository provided by the Google Code issue tracking system.

2.3.4. Libraries

The wiiverse library: [11] Wiiuse is a library written in C that connects with several Nintendo Wii remotes. Supports motion sensing, IR tracking, nunchuk, classic controller, and the Guitar Hero 3 controller. Single threaded and nonblocking makes a light weight and clean API. The wiiverse library is used in Watch & Touch as a layer of abstraction between the IWBC and the Wiimote; it handles the Bluetooth communication between the Wiimote and the instructor’s machine and exposes a number of methods for accessing the Wiimote's functionality without going into the details of the communication protocol.
The poppler library: [10] Poppler (or libpoppler) is a free software library used to render PDF documents. It is used by the PDF viewers of the open source GNOME and KDE desktop environments, and its development is supported by freedesktop.org. The project was started by Kristian Høgsberg with two goals in mind: To provide PDF rendering functionality as a shared library, in order to centralize maintenance effort, and to go beyond the goals of Xpdf, and integrate with functionality provided by modern operating systems. Poppler itself is a fork of the Xpdf-3.0 PDF viewer developed by Derek Noonburg of Glyph and Cog, LLC. Watch & Touch uses the libpoppler Qt4 interface, which allows rendering of PDF files directly at the desired resolution (as well as numerous other operations) from inside Qt to display PDF content.

3. Configuration Management Process

3.1. Identification

The identification of the items subject to the Configuration Management Process are crucial towards efficient construction and application of the plan. These items, split into three categories, are given below:

3.1.1. Hardware Components

The hardware components utilized in Watch & Touch are quite important towards building a successful implementation of the project. Although their acquisition and construction are relatively simple, the low-cost constraint and stability of usage make it necessary to further investigate how the existing hardware components could be improved. These components are listed as follows:

- **Wiimote**: The hardware controller designed by Nintendo for use with the Nintendo Wii gaming console, that offers the tracking of movements of up to 4 infrared points. Both in terms of its performance and pricing, the Wiimote is an excellent solution for Watch & Touch and forms the hardware backbone of the project.
- **Wiimote holder**: The Wiimote has to be positioned in a certain manner with respect to the projection surface. The Wiimote holder provides this physical positioning. The current holders in use are constructed from metal tape and a mini camera tripod and should be sufficiently functional for the execution of the project.
- **IR Sources**: At least one IR-light emitting device is necessary to take advantage of the interactive capabilities provided by Watch & Touch. Currently the team is utilizing hand-constructed button-battery-LED devices as IR sources, which are functional but unfortunately are prone to failure due to the low quality of electronic components used in their construction.
3.1.2. Source Code

As a software project, the process of source code creation and manipulation is the most important aspect of project development. We utilize the modular decomposition provided in the Watch & Touch SDD [3] to identify configuration items of the CMP:

- **ContentDisplay Module** - The ContentDisplay module provides the necessary functionality for displaying the three primary kinds of visual content the IWBC supports: presentations, videos and webpages.
- **Sessions Module** - The Drawing module contains the components responsible for realizing the drawing operations which play a prime part in Watch & Touch’s functionality.
- **Drawing Module** - The Sessions module includes the components related to the administration and functioning of the collaborative drawing sessions.
- **WiimoteInput Module** - The WiimoteInput module forms the core of the IR‐pen input functionality of the IWBC; it communicates with the Wiimote via the wiiuse library to receive the changes of the IR point position.
- **Configuration Module** - The Configuration module is responsible for keeping the user preferences regarding the functioning of the IWBC.
- **Task Management Module and User Interfaces** - The TaskManagement module is responsible for the logical and visual management of the instances of multiple tasks which the IWBC is capable of - creating new task instances, switching between and terminating existing instances, and forming a basis for the user interface elements related with these tasks. The user interfaces themselves are created with QML [5] which further abstracts the interfaces from the business logic.

3.1.3. Documentation

Aside from the obligatory documentation of software requirements and design details, documentation plays a particularly important role for Watch & Touch. As it targets a group of users who are not necessarily tech-savvy or accustomed to interactive whiteboards, the user manual is also CI or great importance towards achieving Watch & Touch’s goals.

- Project Proposal
- Requirements Analysis Report
- Initial Design Report
- Detailed Design Report
- Configuration Management Plan
- Web page
- Blog entries
- Weekly Reports
- User manual
- Written and Video Tutorials of the final product

9
3.2. Configuration Management and Control

The system change process will be handled differently for different types of CIs.

- For hardware components, any proposed major changes are to be discussed directly in face-to-face team meetings as the hardware resources available to DialecTech are limited and major changes to the existing ones could significantly affect the course of the project. Minor/rudimentary changes or maintenance of hardware (such as strengthening the Wiimote holders, changing the IR pen or Wiimote batteries and so on) can be carried out without requiring any explicit discussion.
- For documentation, since the creation of all documents will take place in Google Docs which offers online collaborative typing and revision tracking, any changes can be made directly to the document by any team member, which can later be reverted if needed during the revision of the document before the final version is submitted.
- For source code, a more strict approach is needed, which is described in sections 3.2.1, 3.2.2 and 3.2.3.

3.2.1. Software System Change Requests

Minor SCR for the software, such as small refactorings or bugfixes, are to be controlled by SVN's revision system and do not need to go through a request-evaluate-implement procedure. However, for every commit to the repository an explanatory log message which should contain an issue number referring to the relevant ticket ID should be included, in the following format:

```
issue #TICKET_ID : description of changes in this commit
```

Major SCRs are controlled by the Google Code issue tracking system. In this system a SCR consists of:

- Team member name
- Description
- Date
- Type (defect, enhancement, feature request, review request, milestone definition..)
- Related Module(s)
- Priority
- Milestone label, if applicable

When a SCR reported a ticked is opened by the issue tracking system and can be seen by all the team members. Care should be taken to include as much information as possible in the Description section to facilitate proper evaluation of the change request by other team members. The progress of any unclear SCRs will be frozen by the project manager until the issuing team member has provided a sufficient level of detail on the request.
3.2.2. System Change Evaluation

The discussions about evaluation of SCR are maintained on tickets in the Google Code issue tracking system. Some higher-level SCRs requiring a lot of change may be discussed during face-to-face meetings and on the team mailing list as well, but any decisions or keypoints on the SCR must be entered to the issue tracking system so as to keep organized reference. During the evaluation, each member can stress his/her opinions freely and the evaluation will be based on these opinions weighed against the goals of the project and the status in development.

3.2.3. System Change Implementation

If SCR is approved after an evaluation, all the possibly affected CI’s will be detected and the related changes will be applied based on detected CI’s. Then the updated version of the CI(s) will be updated by committing the related resources through SVN, in accordance with the log message format specified in section 3.2.1.

3.3. Configuration Status Accounting

The configuration items have been introduced in previous sections and a system for managing software changes was given. The information about related configuration items during changes has to be stored as the control of configuration items will become more difficult as more changes occur during the software development process. Since accounting for the development process of the project is essential, different methods will be used to express the details of these changes. This information will not only improve the inter-team communication but also with the external individuals or organizations might have an interest in following the project. The aforementioned information includes the following:

- configuration item identifications
- information on the change request
- information related to the details of the implementation

As the project approaches its final, we will keep information about the details of configuration items in the SVN commit logs. The meeting reports and weekly progress reports should also provide readable high-level summaries of the recent changes.

Since the SVN commit logs are the most essential unit of information to be provided during changes, care should be taken by developers to ensure the following while committing:

- Commits should not contain two unrelated updates or changes.
- Everyone that has taken part in the commit should be mentioned in the log of the commit.
- Commit comments should describe the change made briefly in a way that everyone can understand.
- Related updates’ commit to a ticket in the issue tracking system should contain the ticket id in the log, in the format specified in section 3.2.1
DialecTech members will also use the team’s blog [6] in order to post entries related to development process and the status of the project. These blog posts may include tutorials or code samples about specific or general matters, approaches or workarounds for issues and forecasts for development status.

3.4. Auditing

Auditing will be accomplished by all the members of DialecTech. During the auditing phase of SCRs, the changes that are made on a CI (or several) will be checked to evaluate the correctness of the changes. Auditing of the source code will be done at the latest in the weekly meetings, but preferably as soon as the changes are made, by using appropriate testing methods. Developers are responsible for determining whether their commits are working correctly. This is a personal responsibility of component developers and should be done before each commit into the SVN repository. Each commit should at least be compiled correctly. By this policy, the source code in the repository is always trusted to be compiled and working. Besides these functional audits, members should observe the physical and process status each week. Project schedule should be checked and updated regularly in order to obey the timing constraints. Number of open tickets and resolved ones should also be evaluated in order to maintain sustainability.

4. Project Schedules and CM Milestones

4.1 Project Schedules

DialecTech follows the Gantt Chart provided in the Watch & Touch SDD [3] for scheduling development. The first Gantt Chart that is about first term has already been finished and for the second one that is about second and current term is being followed. Weekly tasks are re-determined at the weekly meetings to ensure the general schedule is met, which can cause deviations from the work items proposed in the Gantt chart for a particular time.

4.2 CM Milestones

In order to have a healthy development process, the team will be keeping milestones as small as possible by matching them with the demanded deliverables of CENG490. Our main target in this decision is the ability to tolerate possible changes and problems in project schedule. The future milestones and their targets are specified in the following tables, as well as the current situation towards their development. For each milestone the target elements are separated and they will be developed in parallel.
**First Development Snapshot, Demo - due Mar 29, 2011**

<table>
<thead>
<tr>
<th>Milestone Targets</th>
<th>Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two person version of the collaborative drawing client</td>
<td>In progress</td>
</tr>
<tr>
<td>Basic multitouch gestures</td>
<td>In progress</td>
</tr>
<tr>
<td>Screen capture mode support</td>
<td>In progress</td>
</tr>
<tr>
<td>Screencasting support</td>
<td>In progress</td>
</tr>
<tr>
<td><strong>Drawing tools</strong></td>
<td><strong>DONE</strong></td>
</tr>
<tr>
<td><strong>Threaded and cached loading for PDF files</strong></td>
<td><strong>DONE</strong></td>
</tr>
<tr>
<td><strong>Presentation Annotation</strong></td>
<td><strong>DONE</strong></td>
</tr>
<tr>
<td><strong>PPT viewing</strong></td>
<td><strong>DONE</strong></td>
</tr>
<tr>
<td><strong>Google docs support</strong></td>
<td><strong>DONE</strong></td>
</tr>
</tbody>
</table>

**First Release, Demo - May 10, 2011**  
**Time After Previous Milestone: 6 weeks**

<table>
<thead>
<tr>
<th>Milestone Targets</th>
<th>Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some QML interfaces</td>
<td>Waiting</td>
</tr>
<tr>
<td>Full multitouch gestures</td>
<td>Waiting</td>
</tr>
<tr>
<td>Configuration module</td>
<td>Waiting</td>
</tr>
<tr>
<td>Webpage annotation</td>
<td>Waiting</td>
</tr>
<tr>
<td>Video annotation</td>
<td>Waiting</td>
</tr>
<tr>
<td>Collaborative drawing with multiple clients</td>
<td>Waiting</td>
</tr>
<tr>
<td>Testing on Windows systems</td>
<td>Waiting</td>
</tr>
</tbody>
</table>
5. Project Resources

Watch & Touch project resources can be classified in three categories: the financial budget for hardware components, manpower in which human resources of the project are considered and the software resources in which the software tools used for development and execution of the development processes are mentioned.

5.1 Budget

In our project, we envision a solution adhering to the principles of free and open source software, which will bring the ability to utilize educational technology in all classrooms, for the low cost of its off-the-shelf available hardware components. In addition to proposing a better software system, our project’s main difference from the market is being low-cost, which accounts for the project’s low budget.

During our project development, we have purchased 2 Nintendo Wiimote controllers, one of which costs around 75-80 TL. Other expenses that we have covered are the necessary hardware for the execution of the project such as the materials for the Wiimoteholder, IR
pen and the IR rings. The materials for the construction of two Wiimote holders, 1 IR pen and 2 IR rings cost us around 20 TL in total. The tablet devices needed to test the CBC components are personal property of the developers and do not come at any extra specific cost. Hence, up to now, our total expenses were around 180 TL which can be considered quite affordable. This was evenly divided among the team members and paid.

5.2 Manpower
The manpower is provided only by our team members. As Watch & Touch is being constructed as a CENG490 Senior Design course project, all DialecTech team members work voluntarily and without expecting any payment for their labor.

5.3 Software Tools
The following tools are being / will be used during the development process, either in the software development itself or for project management and team communication tasks.

- SVN
- Google Code issue tracking system
- Google Documents
- Google Groups
- Qt Creator
- Wordpress

6. Plan Optimization
During the Watch & Touch software development process, it has been crucial to obey CMP for plan optimization. As task distribution, that is shown in living schedule, is done according to the interest, excellence and experience of the team and each member of DialecTech is involved in the project, plan optimization has already been managed. Living schedule will be followed regularly. However, as the project is ongoing, it would be necessary to make some changes in the deadlines of the living schedule during the development process. Therefore, our living schedule is organized flexible enough to tolerate these changes. These changes will be done in pursuance of change management process. Changes and other optimizations will be decided by team in weekly meetings and updated in the living schedule, Gantt chart and CMT. Updated schedule will be reached via project web site.