FHZ → FACHHOCHSCHULE ZENTRALSCHWEIZ

HTA → HOCHSCHULE FÜR TECHNIK+ARCHITEKTUR LUZERN

Abteilung Informatik

Java

Thread Skalierung

SCMP

Software Configuration Management Plan

HTA Horw

HTA LUZERN
Technikumstrasse 21
CH—6048 Horw

T: 041-349-33-11
F: 041-349-39-60
W: www.hta.fhz.ch

Diplomarbeit 2006
Aregger Marcel
Meier Rainer

Änderungskontrolle

Version	Datum	Ausführende Stelle	Bemerkungen/Art der Änderung
1.1	2006-10-16	Rainer Meier	Initial Release
1.2	2006-10-20	Rainer Meier	Vorlagen eingefügt

Prüfung und Freigabe

Vorname/Name	Dokumentversion	Status	Datum	Visum
Rainer Meier	1.2	Final	2006-10-20	
Marcel Aregger	1.2	Final	2006-10-20	

1. Inhaltsverzeichnis

2. Einleitung

Dieses Dokument richtet sich nach dem Standard IEEE 828-1990.

2.1. Referenzierte Dokumente

Tabelle 1 Referenzierte Dokumente

Referenz	Beschreibung
[1]	SPMP, Software Project Management Plan

2.2. Definitionen und Abkürzungen

Tabelle 2 Abkürzungen

Abkürzung	Beschreibung
SPMP	Software Project Management Plan
SCMP	Software Configuration Management Plan
CI	Configuration Item
CCM	Change & Configuration Management
CVS	Concurrent Versioning System
IEEE	Institute of Electrical and Electronic Engineers
SPMP	Software Project Management Plan
SCMP	Software Configuration Management Plan

2.3. Links

Tabelle 3 Links

Referenz	Beschreibung
[CVS]	Concurrent Versioning System (CVS), http://www.cvshome.org/

3. Change & Configuration Management

Diese Kapitel beschreibt, wie das Change & Configuration Management gehandhabt wird. Dies umschliesst sowohl Dokumente als auch Software bzw. Sourcecode.

3.1. Organisation

In [1] werden die Verantwortlichkeiten im Projekt definiert. Dort wird eine Person als verantwortlicher für die CCM-Rolle definiert. Diese Rolle ist verantwortlich für das Qualitätsmanagement auf der Ebene der Versionskontrolle und Verwaltung.

Ebenso wird in [1] Rolle für Finanzen und Controlling definiert. Diese Rolle übernimmt das Zeitmanagement, die Kontrolle des Projektfortschrittes und der eingesetzten Mittel.

3.2. Verantwortungen

Die Verantwortlichen für das Change & Configuration Management werden in [1] genannt. Die aufgaben sind wie folgt definiert:

- Definition eines Configuration Management Planes (dieses Dokument)
- Festlegung von Namenskonventionen
- Festlegung eines Versionisierungskonzeptes
- Technische Umsetzung und Bereitstellung der benötigten Tools

4. Aktivitäten

Dieses Kapitel beschreibt wie CI's identifiziert und behandelt werden. Da es sich hier um ein kleineres Projekt handelt wurden die Aktivitäten nicht aufgeteilt in Identifizierung und Kontrolle. Die folgenden Punkte beschreiben sowohl die Identifizierung als auch die Kontrolle des Umgangs mit den Dokumenten/Dateien (im Folgenden CI's genannt).

4.1. Feststellen von Konfigurations-Elementen (CI's)

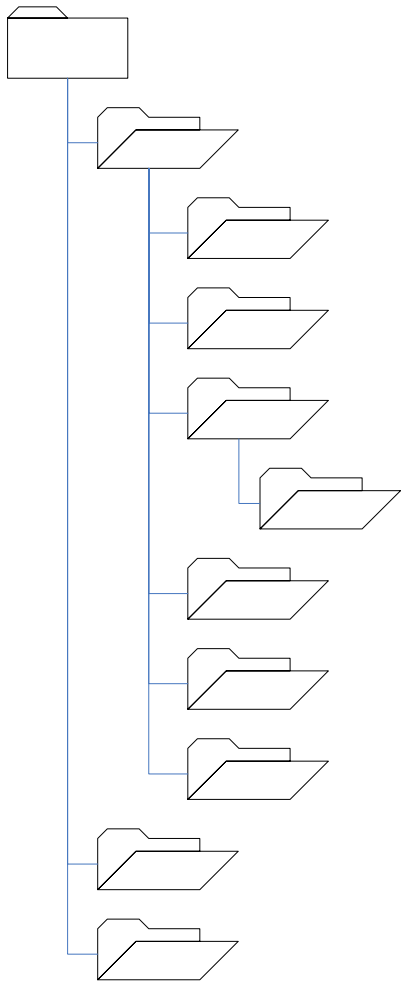
Konfigurations-Elemente (CI's) werden von der CCM-Rolle definiert. Die verantwortliche Person identifiziert Fragmente (Dokumente, Sourcecode usw.) die versionisiert werden sollen. Projektmitarbeiter können weitere CI's vorschlagen. Dies kann durch eine unformelle Anfrage geschehen (mündlich/schriftlich). In jedem Fall soll CCM aber die Ablage und korrekte Benennung prüfen.

4.2. Benennung von Konfigurations-Elementen (CI's)

Alle CI's werden im CVS (siehe [CVS]) abgelegt und automatisch versionisiert. CCM ist dafür verantwortlich die dazu notwendige Server-Infrastruktur zur Verfügung zu stellen.

Für die finale Abgabe werden von allen Dokumenten PDF-Dokumente erzeugt.

Die Verzeichnisstruktur sieht wie folgt aus:



Dokumente	Dokumentablage, die Dokumentation des Projektes wird hier abgelegt.
Allgemein	Allgemeine Dokumente wie der Projektauftrag, die Definition des Posters, allgemeine Regelungen und Bestimmungen.
Projektplan	Projektplanung (Zeitplan)
SPMP	Software Project Management Plan
source	Originalquellen, die in den Dokumenten verwendet werden
SCMP	Software Configuration Management Plan
Analyse	Eigentliche Projekt-Dokumentationen wie die Basisanalyse, Evaluierung, Umsetzung, Test und Auswertung
...	Weitere Dokumentationen
<projekt>	Source Code für Software Projekt
...	Source Code für weiteres Software Projekt

Abbildung 1 CVS Verzeichnisstruktur

Die Dokumente in den Verzeichnissen tragen eine eindeutige Bezeichnung wie „SPMP“ oder „SCMP“ und ihre Datei-Erweiterung wie „.doc“ oder „.pdf“. Insbesondere werden keine Erweiterungen wie Datum oder Versionsnummer eingefügt. Die Versionisierung übernimmt CVS. Ältere Versionen sind jederzeit aus CVS abrufbar.

Häufig beinhalten die Dokumente Zeichnungen oder Bilder, die in anderen Programmen erstellt werden. Die Quelldateien dieser Fragmente können jeweils im Unterordner „source“ abgelegt werden. Auch hier gilt die oben genannte Namenskonvention.

Vorlagen für Source Code und Dokumente sind ebenfalls im CVS verfügbar. Siehe dazu auch Kapitel 6.

4.3. Bearbeitung von Konfigurations-Elementen

CVS erlaubt die gleichzeitige (konkurrierende) Bearbeitung von Dateien. Dies ist bei der Bearbeitung von Source-Code (Plain-Text) sehr hilfreich da mehrere Benutzer gleichzeitig an einer Datei arbeiten können. Das überschreiben von Änderungen anderer Benutzer wird von CVS automatisch verhindert. CVS hilft dann auch beim zusammenführen (Merge) mehrerer Versionen.

Leider funktioniert dies prinzipiell nicht mit binären Datenstrukturen. Unglücklicherweise sind MS Office Dokumente aber binär abgespeichert und können so nicht auf einfache Art und Weise zusammengeführt werden.

Somit ist es ratsam, dass nur immer ein Benutzer an einem Dokument arbeitet. Da in diesem Projekt nur sehr weniger Mitarbeiter an den Dokumenten arbeiten lässt es sich leicht absprechen wer an welchem Dokument arbeitet.

Sollten die Dokumente trotzdem zeitgleich von mehreren Benutzern bearbeitet werden bietet Word im Menü „Extras => Dokumente Vergleichen und zusammenführen…“ bzw. „Tools => Compare and Merge Documents…“ eine Funktion um die Zusammenführung zu erleichtern.

Fragen bezüglich CVS und der Verwendung kann der definierte Change & Configuration Manager (CCM) beantworten.

4.4. Konfigurations-Status

Im CVS eingecheckte Dokumente gelten prinzipiell als freigegeben zur Weiterbearbeitung. Bei Veränderungen der Struktur oder sonstigen gravierenden Änderungen ist der Change & Configuration Manager (CCM) zu informieren.

Sollen die CI’s auf einem bestimmten Status (z.B. „known-good“ Status) eingefroren werden, so kann der CCM ein so genanntes „Tag“ (auch „Label“ genannt) auf alle CI’s legen. Dieser Status kann dann zu einem späteren Zeitpunkt problemlos wiederhergestellt werden. Insbesondere bei der Entwicklung von Sourcecode sind freigegebene Versionen über ein Tag zu kennzeichnen.

5. Ressourcen

CCM-Arbeiten laufend erledigt. Eine fixe Ressourcenvergabe ist nicht geplant. Die Versionisierung und Freigabe von Dokumenten ist ein zentrales Element der Qualitätssicherung und darf auf keinen Fall vernachlässigt werden. Aus diesem Grunde ist bei jeder Aktion die zur Veränderung/Erstellung von Dokumenten führt genügend Zeit für das Versionisierungs-Management einzuplanen.

Bei Fragen zur Versionisierung ist immer der verantwortliche CCM zu kontaktieren.

6. Anhang

6.1. Java Source Code Vorlage

```
/*
 * Created on 20.10.2006
 */
package ch.skybeam.benchmark;

/**
 * [class comment - for what is it used, special cases etc.]
 *
 * <h3>History:</h3>
 * <code>
 * <br>$Log:  $
 * </code>
 *
 * @author $Author: $
 * @version $Revision: $
 */
public class Sample {
    /** RCS Header - to identify PATH and version */
    public static final String rcsHeader = "$Header:  $";

    /** RCS Name - to identify Label */
    public static final String rcsName = "$Name:  $";

    /** RCS Revision - to identify version numer */
    public static final String rcsRevision = "$Revision:  $";
}
```

6.2. Java Interface Vorlage

```
/*
 * Created on 20.10.2006
 */
package ch.skybeam.benchmark;

/**
 * [class comment - for what is it used, special cases etc.]
 *
 * <h3>History:</h3>
 * <code>
 * <br>$Log:  $
 * </code>
 *
 * @author $Author: $
 * @version $Revision: $
 */
public interface Sample {
    /** RCS Header - to identify PATH and version */
    public static final String rcsHeader = "$Header:  $";

    /** RCS Name - to identify Label */
    public static final String rcsName = "$Name:  $";

    /** RCS Revision - to identify version numer */
    public static final String rcsRevision = "$Revision:  $";
}
```

6.3. C++ Source Code / Header Vorlage

```
/**************************************************************************
Created on:  20.10.2006

Type:        C++ Source

Description: [class comment - for what is it used, special cases etc.]

Author:      $Author:  $
Revision:    $Revision:  $

Revision history:
$Log:  $

**************************************************************************/

#include <stdio>
```

7. Verzeichnisse

7.1. Tabellenverzeichnis

7.2. Abbildungsverzeichnis

7.3. Index