Zero1one

Dokumentation von Gruppe 2

vorgelegt am 28. Januar 2020

Fakultät Wirtschaft

Studiengang Wirtschaftsinformatik

Kurs [Kursbezeichnung]

von

[Vorname Nachname]

|  |  |
| --- | --- |
| Betreuer in der Ausbildungsstätte: | DHBW Stuttgart: |
| [Name des Unternehmens] [Titel, Vorname und Nachname d. Betreuers] [Funktion des Betreuers] [Unterschrift des Betreuers] | [Titel, Vorname und Nachname des wissenschaftl. Betreuers/Prüfers] |

**Inhaltsverzeichnis**

[Abkürzungsverzeichnis (bei Bedarf) III](#_Toc53756011)

[Abbildungsverzeichnis (bei Bedarf) IV](#_Toc53756012)

[Tabellenverzeichnis (bei Bedarf) V](#_Toc53756013)

[1 Einleitung 1](#_Toc53756014)

[1.1 Anforderungsanalyse 2](#_Toc53756015)

[1.2 Pflichtenheft 3](#_Toc53756016)

[2 Projektorganisation und Projektphasen 4](#_Toc53756017)

[2.1 Projektorganisation 4](#_Toc53756018)

[2.2 Projektphasen 5](#_Toc53756019)

[3 Nächstes Kapitel 6](#_Toc53756020)

[3.1 Abschnitt 6](#_Toc53756021)

[3.1.1 Unterabschnitt 6](#_Toc53756022)

[3.1.2 Unterabschnitt 6](#_Toc53756023)

[3.2 Entsprechend 6](#_Toc53756024)

[3.2.1 Fortgesetzt 6](#_Toc53756025)

[3.2.2 Nochmal fortgesetzt 6](#_Toc53756026)

[4 Schlussteil 7](#_Toc53756027)

[1.1.1 Anhang (bei Bedarf) 8](#_Toc53756028)

[Literaturverzeichnis 10](#_Toc53756029)

# Abkürzungsverzeichnis (bei Bedarf)

CMS = Configuration Management System

CI = Configuration Item

ITIL = Information Technology Infrastructure Library

# Abbildungsverzeichnis (bei Bedarf)

[Abb. 1: Eine Abbildung 2](#_Toc31092940)

# Tabellenverzeichnis (bei Bedarf)

[Tab. 1: Eine Beispieltabelle 3](#_Toc31092943)

1. Einleitung

Durch die Information Technology Infrastructure Library (ITIL), welche als eines der meist genutzten Frameworks für das IT Service Management gilt,[[1]](#footnote-1) werden verschiedene Konzepte und Leitlinien zur praktischen Handhabung des IT Service Managements zur Verfügung gestellt.[[2]](#footnote-2) Einer der in ITIL beschriebener Prozesse stellt hierbei das Configuration Management dar.

Das Configuration Management dient der Handhabung von IT-Komponenten, wobei sichergestellt werden, dass die Assets ordentlich kontrolliert werden und dass korrekte Informationen über diese verfügbar sind.[[3]](#footnote-3) Das Ziel ist somit die Bereitstellung einer zentralen Informationsquelle über alle IT-Komponenten und Ressourcen und deren Abhängigkeiten untereinander. Hierfür werden Informationen vor allem über die Konfiguration der Assets und die Beziehung zwischen den einzelnen Assets in einem Configuration Management System (CMS) dokumentiert und in einer dazugehörigen Configuration Management Database abgespeichert. Die einzelnen Assets und deren Konfigurationsdetails werden hierbei anhand von Configuration Records festgehalten. [[4]](#footnote-4)

Die einzelnen Assets oder Komponenten werden als Configuration Item (CI) bezeichnet. Die unterschiedlichen CIs können hierbei stark in Komplexität, Größe und Art variieren. So kann ein CI ein komplettes System inklusive aller Hardware, Software und Dokumentationen oder auch nur ein kleine Hardwarekomponente darstellen.[[5]](#footnote-5) Um die vielen unterschiedlichen CIs besser zu strukturieren, werden unterschiedliche Configuration Item Typen definiert, welche üblicherweise alle eingesetzten Hardware und Software Komponenten, wie zum Beispiel Server oder Datenbanken, beinhalten. Jedes CI und der dazugehörige Configuration Record ist somit genau einem Configuration Item Typ zugeordnet.[[6]](#footnote-6) Die CIs werden anhand von Configuration Records, welche alle für das bestimmte CI relevanten Informationen beinhalten, in einem CMS festgehalten. Hierbei beschreibt jeder Configuration Record genau ein bestimmtes CI, wobei jedes CI ein eindeutiges Kennzeichen haben muss.[[7]](#footnote-7)

In der nachfolgenden Dokumentation ist nun detailliert der Entwurf, die Entwicklung und die anschließende Qualitätssicherung und Wartbarkeit eines CMS beschrieben. Dabei werden zunächst in den nachfolgenden Abschnitten die genauen funktionalen und nicht funktionalen Anforderungen von der Seite des Auftraggebers an das CMS betrachtet und anschließend deren Umsetzung durch das Projektteam dargestellt.

* 1. Anforderungsanalyse

Nach der groben Beschreibung des Auftrags sollen nun die konkreten funktionalen und anschließend die nicht funktionalen Anforderungen an das CMS betrachtet werden.

Mithilfe des CMS sollen alle möglichen verschiedenen CI-Typen verwaltet werden können. Bei der Erstellung eines neuen CI-Typs sollen auch für diesen Typ übliche Attribute erstellt werden können, sodass nicht für jeden CI-Record die Attribute einzeln erstellt werden müssen, sondern bereits die vordefinierten Attribute des jeweiligen Typs vorhanden sind. Außerdem sollen die CI-Typen auch gelöscht werden können, wenn diese nicht mehr benötigt werden, um die Übersichtlichkeit des Systems zu erhöhen.

Um die einzelnen CIs im CMS festzuhalten soll für jedes CI ein CI-Record mit einem bestimmten CI-Typ einfach erstellt werden können. Hierbei sollen neben den bereits vordefinierten Attributen noch weitere hinzugefügt werden können, um auch alle besonderen Konfigurationen eines speziellen CIs im zugehörigen CI-Record abzuspeichern. Des Weiteren sollen die CI-Records auch im Nachhinein noch änderbar sein, damit die CI-Records bei Neuerungen schnell aktualisiert werden können und nicht neu erstellt werden müssen. Da viele CIs nach bestimmter Zeit auch ausgetauscht werden, sollen die zugehörigen CI-Records im CMS gelöscht werden können. Außerdem sollen die CI-Records einsehbar und möglichst schnell auffindbar sein, sodass man die gespeicherten Informationen möglichst schnell und einfach finden kann.

Um die Übersichtlichkeit noch weiter zu erhöhen, soll angezeigt werden, wie viele CI-Records zu einem bestimmten CI-Typ vorhanden sind. Wenn möglich sollten auch Diagramme in Form von Kreis oder Balkendiagrammen zur Veranschaulichung der Mengenverteilung erstellt werden.

Eine weitere angeforderte Funktionalität stellt das Login dar. Hierbei soll der Benutzer sich mit einem Benutzernamen und einem Passwort anmelden können und bei den richtigen Eingaben Zugang zur Software erlangen. Je nach der Berechtigung des angemeldeten Benutzers kann dieser dann unterschiedliche Tätigkeiten durchführen. Nach Beendigung der Tätigkeiten soll der Benutzer sich wieder ausloggen können, damit die Sitzung beendet wird und niemand Unberechtigtes Zugang erhält.

Die Benutzer sollen in unterschiedliche Benutzergruppen unterschieden werden, sodass verschiedene Berechtigungen vergeben werden können. Eine Benutzergruppe soll hierbei berechtigt sein, neue Benutzer anzulegen, wobei der Benutzername festgelegt und ein initiales Passwort vergeben wird, sodass der Benutzer sich erstmal mit dem vorher festgelegten Passwort anmelden kann und dieses dann anschließend ändern kann. Bei der Erstellung des Benutzers soll auch die dem Benutzer zugeordnete Berechtigung vergeben werden. Diese Rechte sollen auch später noch bearbeitbar sein und gegebenenfalls der Benutzername abgeändert werden können. Außerdem sollen Benutzer auch gelöscht werden können.

Neben den funktionalen Anforderungen gibt es auch noch die nicht funktionalen Anforderungen an das CMS, welche nun genauer betrachtet werden sollen. Dabei soll für eine möglichst gute Bedienbarkeit gesorgt werden, sodass die Benutzung möglichst intuitiv erfolgen kann und nicht anhand einer komplizierten Anleitung befolgt werden muss. Eine weitere Eigenschaft, wie die Bedienbarkeit verbessert werden kann, ist die Plausibilitätsprüfung bei den Eingaben. Des Weiteren soll die Performance des Systems optimiert werden, indem zum Beispiel die Antwortzeiten möglichst geringgehalten werden. Außerdem soll die Software wartbar und auch änderbar sein, damit diese in Zukunft angepasst werden kann, wenn es Änderungsbedarf gibt. Eine wichtige Anforderung an das System stellt auch die Sicherheit dar, da das CMS viele wichtige Daten beinhaltet, welche nur durch Berechtigte anpassbar sein sollen und nicht durch Unberechtigte manipuliert werden können.

* 1. Pflichtenheft

1. Projektorganisation und Projektphasen
   1. Projektorganisation

Um ein Projekt erfolgreich durchzuführen gilt es zuerst einmal eine Projektorganisation aufzustellen. Hierbei unterscheidet man grundsätzlich zwischen drei Formen der Projektorganisation:

* Reine Projektorganisation
* Stabs Projektorganisation
* Matrix Projektorganisation

Das Ziel einer Organisationsstruktur ist es das gemeinsame Rollenverständnis innerhalb des Projektteams zu unterstützen und für klare Verantwortlichkeiten und Befugnisse aller Projektmitarbeiter zu sorgen, sodass das Projekt schneller und reibungsloser verlaufen kann.[[8]](#footnote-8)

Da es sich bei der Fallstudie zum Thema Softwareentwicklung um ein Projekt handelt, welches neben dem alltäglichen Vorlesungsgeschehen bearbeitet werden soll, fiel die Wahl der Projektorganisation auf eine Stabs Projektorganisation. Bei der Stabs-Projektorganisation hat der Projektleiter, welcher bei uns durch Daniel Reimann übernommen wurde, eine reine koordinierende Funktion, also keine Weisungsrechte. Der Projektleiter handelt als Auftraggeber für die anderen Abteilungen, die an dem Projekt mitarbeiten.

Um die Arbeit an der Fallstudie zu erleichtern wurde die Gruppe in zwei Untergruppen unterteilt, einmal die Gruppe Frontend und die Gruppe Backend, diese können wie einzelne Abteilungen angesehen werden. Zudem hat Marcel Mahlfeld neben seiner Mitarbeit in der Abteilung Frontend noch die Organisation und Koordination der Projektdokumentation übernommen.

Die beiden Gruppen wurden nach Interessen und vor allem nach dem Wissen eingeteilt, so haben sich die jeweiligen Gruppenmitglieder die mit dem Backend schon Erfahrungen haben zusammengeschlossen, genauso wie die Mitglieder die mit dem Frontend bereits Erfahrungen haben. Damit wurde bezweckt, dass die jeweiligen Gruppen so eingeteilt sind, dass jedes Gruppenmitglied etwas zum Projekt beitragen kann und seinen eventuellen Stärken bestmöglich genutzt werden können. Somit wird die Zusammenarbeit innerhalb der Gruppen erleichtert und das Projekt kann schneller und besser zum gewünschten Ziel geführt werden.

Auch innerhalb der einzelnen Untergruppen wurden die verschiedenen Aufgaben weiter unterteilt und unter den Gruppenmitgliedern aufgeteilt, somit sollte eine schnelle Erreichung des Gesamtzieles ermöglicht werden. Durch die Verteilung der einzelnen Aufgaben auf die verschiedenen Gruppenmitgliedern wurden die einzelnen Personen entlastet und die Aufgaben konnten parallel zueinander erledigt werden.

Damit alle Mitglieder des Projektes auf dem momentanen Stand des Projektes sind wurden wöchentliche Termine vereinbart bei denen dann die Untergruppen ihren aktuellen Stand präsentierten und einen groben zeitlichen Überblick lieferten wie lange sie noch ungefähr benötigen. Zudem wurden aufgetretene Probleme und Hindernisse in den wöchentlichen Treffen besprochen, die dazugehörige Lösung wurde dann ebenfalls erläutert. Sollte noch keine Lösung gefunden worden sein konnte nun in der großen Gruppe nach einer Lösung gesucht werden, welche meist auch relativ schnell gefunden wurde. Um im Nachhinein noch nachvollziehen zu können was an den einzelnen Terminen besprochen wurde, hat der Daniel Reimann bei jedem Treffen ein Protokoll geführt, in dem die einzelnen Punkte die besprochen wurden festgehalten sind. Dieses Protokoll steht jedem Projektmitglied zur Verfügung, sodass jeder Zeit die einzelnen Projektmitglieder noch einmal nachschauen können was genau besprochen wurde. Durch das Protokoll hatten auch Projektmitglieder die ein Treffen verpasst hatten die Möglichkeit sich über das verpasste zu informieren ohne eins der anderen Projektmitglieder fragen zu müssen.

Auch innerhalb der Untergruppen wurden Treffen vereinbart welche oftmals zusätzliche zu den wöchentlichen Treffen waren um dort dann individuell Themen zu bearbeiten und Probleme so wie Hindernisse aus der Welt zu schaffen, welche oftmals nur die jeweilige Untergruppe betreffen.

* 1. Projektphasen

Für den erfolgreichen Abschluss eines Projektes sollte dieses zunächst einmal in verschiedene Phasen unterteilt werden, hierdurch soll mehr Übersichtlichkeit innerhalb des Projektes geschaffen werden. Zudem wird durch die Unterteilung die Organisation des Projektes vereinfacht, da somit ein zeitlicher Ablauf bereits grob vorgegeben ist und die Einteilung, der für die einzelnen Phase, benötigten Zeit erleichtert wird.

Wir haben unser Projekt in fünf Phasen unterteilt:

1. Die Organisations- und Programmentwurfsphase
2. Die Erstellungsphase
3. Die Zusammenführungs- und Anpassungsphase
4. Die Test- und Dokumentierungsphase
5. Die Fertigstellungsphase

Parallel zu diesen fünf Phasen wurde zudem noch die Projektdokumentation von allen am Projekt beteiligten Personen verfasst.

* + 1. Die Organisations- und Programmentwurfsphase

Bevor die eigentliche Arbeit an dem Projekt gestartet werden konnte galt es zu erst einmal die Projektorganisation zu erarbeiten. Sobald die Untergruppen aufgeteilt waren haben sich die einzelnen Gruppen zusammen gesetzt und einen Programmentwurf erstellt, welcher dann in einem Treffen der gesamten Gruppe vorgestellt wurden, um schon im Voraus eventuell auftretende Probleme oder Hindernisse zwischen dem Frontend und dem Backend zu vermeiden.

* + 1. Die Erstellungsphase

Nachdem der Programmentwurf erstellt und besprochen ist geht das Projekt über in die Erstellungsphase. In dieser Phase erledigen die beiden Untergruppen getrennt von einander ihre für das Projekt benötigten Aufgaben. Natürlich bleibt der Austausch von Informationen und Fortschritten, in Hinsicht auf die Aufgaben, zwischen den beiden Gruppen sehr eng und ausführlich. Dies bewirkt das Probleme und Unstimmigkeiten zwischen den Aufgaben des Frontend und des Backend sehr früh schon erkannt werden und somit schnell wieder behoben werden können bevor es zu komplizierteren Problemlösungen kommen muss.

* + 1. Die Zusammenführungs- und Anpassungsphase

Nachdem die beiden Gruppen ihre Aufgaben soweit wie möglich erledigt haben, kommt es nun zur Zusammenführung der Ergebnisse. Hierbei treten kleinere Unstimmigkeiten auf welche dann in Zusammenarbeit der beiden Gruppen meist sehr schnell und effizient gelöst werden können.

* + 1. Die Test- und Dokumentationsphase

Ist das Programm nun fertig zusammengeführt und alle Anpassungen sind erledigt gilt es nun das Programm zu testen. Hierbei geht es darum herauszufinden ob alle Funktionen des Programms so funktionieren wie sich er Auftraggeber es vorstellt. Auch hier Arbeiten die beiden Gruppen wieder zusammen. Ist die Funktionsweise zufriedenstellend überprüft und mögliche Fehler behoben worden, gilt es nun noch die Dokumentation zu verfassen. Für die Dokumentation werden, alle Protokolle und bereits verfasste Teile der Dokumentation, die im Laufe des Projektes verfasst worden sind, zusammengetragen und zu einer Dokumentation vereint. Dadurch entsteht eine möglichst genau Wiedergabe des Projektverlaufs und eine ausführliche Erläuterung der getroffenen Entscheidungen.

* + 1. Die Fertigstellungsphase

In der letzten Phase der Fertigstellungsphase werden noch letzte Änderungen vorgenommen damit die Wünsche des Auftraggebers so gut wie möglich erfüllt werden.

1. Nächstes Kapitel

Text

* 1. Abschnitt

Text

* + 1. Unterabschnitt

Text

* + 1. Unterabschnitt

Text - bitte daran denken, dass jede Gliederungsebene mindestens zwei Elemente umfassen muss.

* 1. Entsprechend

Text

* + 1. Fortgesetzt

Text

* + 1. Nochmal fortgesetzt

Text

1. Schlussteil

### Anhang (bei Bedarf)

[Anhang 1: Beispielüberschrift 5](#_Toc505081892)

[Anhang 2/1: Teil 1 des Anhangs 2 5](#_Toc505081893)

[Anhang 2/2: Teil 2 des Anhangs 2 6](#_Toc505081894)

Anhang 1: Beispielüberschrift

Hier können Sie dann Unterlagen einfügen (Empfehlung: Im Text auf den Anhang verweisen). Die Überschrift des Anhangs bitte über Start – Formatvorlagen – Anhang formatieren; nur dann kann ein korrektes Anhangsinhaltsverzeichnis erstellt werden, das nicht im Inhaltsverzeichnis vorne erscheint.

Anhang 2/1: Teil 1 des Anhangs 2

Bei Anhängen, die nicht auf eine Seite passen, können Sie den Anhang ggf. aufteilen.

Anhang 2/2: Teil 2 des Anhangs 2

Bei Anhängen, die nicht auf eine Seite passen, können Sie den Anhang ggf. aufteilen.

**Erklärung**

Ich versichere hiermit, dass ich meine *[Projektarbeit / Bachelorarbeit / Seminararbeit] mit dem Thema: [Thema einfügen]* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ort, Datum) (Unterschrift)

Literaturverzeichnis

**Gross, J. (2008):** Best Practice im wertorientierten IT-Servicemanagement — zwei Frame-works im Vergleich, in: HMD Praxis der Wirtschaftsinformatik, 45. Jg., Nr. 6, S. 38–47

**Hemmrich, A./Harrant, H. (2016):** Projektmanagement, In 7 Schritten zum Erfolg, 4. Aufl., München: Hanser

**Leimeister, J. M. (2020):** Service Management und Service Operations, in: Dienstleistungsengineering und -management, (Hrsg.: Leimeister, J. M.), 2nd ed. 2020, Berlin, Heidelberg: Springer Berlin Heidelberg, S. 279–342

**Marrone, M./Kolbe, L. M. (2011):** Einfluss von IT-Service-Management-Frameworks auf die IT-Organisation, in: WIRTSCHAFTSINFORMATIK, 53. Jg., Nr. 1, S. 5–19

**Rance, S. (2011):** ITIL, Service Transition, 2011 ed., 2. impr, London: TSO The Stationery Office

1. Vgl. Marrone/Kolbe 2011, S. 5 [↑](#footnote-ref-1)
2. Vgl. Gross 2008, S. 38 [↑](#footnote-ref-2)
3. Vgl. Leimeister 2020, S. 299 ff. [↑](#footnote-ref-3)
4. Vgl. Rance 2011, S. 89 f. [↑](#footnote-ref-4)
5. Vgl. Rance 2011, S. 93 [↑](#footnote-ref-5)
6. Vgl. Rance 2011, S. 308 f. [↑](#footnote-ref-6)
7. Vgl. Rance 2011, S. 94 [↑](#footnote-ref-7)
8. Vgl. Hemmrich/Harrant 2016, S. 24 [↑](#footnote-ref-8)