

IT Projekt Management
WS 2005 - 2006

PS Framework

Nicolas Lanquetin
Matrikel: 0014160

LastChangedDate: 2006-02-19 23:48:05 +0100 (Sun, 19 Feb 2006)
LastChangedRevision: 35

Tools: \LaTeX MiKTeX 2.4, TeXnicCenter 1 Beta 6.31, MS Visio 2003, MS Excel 2003,
MS Project 2003, Dia 0.94, PDFCreator 0.8.0, Subversion 1.3.0

Inhaltsverzeichnis

I. Project Brief	5
1. Zweck und Geltungsbereich	7
2. Hintergrund	8
3. Projektdefinition	9
3.1. Projektziele	9
3.2. Projektumfang	9
3.3. Project Approach	9
3.4. Ausschüsse	11
3.5. Einschränkungen	11
3.6. Voraussetzungen	11
4. PS Business Case	12
5. Qualitätserwartungen	13
6. Abnahmekriterien	14
7. Risiken	16
II. Project Initiation Document	17
1. Zweck und Geltungsbereich	19
2. Hintergrund	20
3. Projektdefinition	21
3.1. Projektziele	21
3.2. Projektumfang	21
3.3. Project Approach	21
3.4. Projekt Lieferergebnisse	23
3.5. Ausschlüsse und Einschränkungen	23
3.6. Voraussetzungen und Annahmen	23
3.7. Externe Abhängigkeiten und Schnittstellen	24

4. Projekt Management Team Struktur und Rollen-Definitionen	25
4.1. Projekt Beteiligte	25
5. Überblick Business Case	26
6. Project Plan	27
6.1. Planübersicht	27
6.1.1. Management Zusammenfassung	27
6.1.2. Planungsvoraussetzungen und -annahmen	27
6.1.3. Externe Abhängigkeiten	28
6.2. Projekt Resultate / Produkte	28
6.2.1. Beschreibung der Hauptprodukte	28
6.2.2. Projekt Gantt Plan (Product Flow Diagram (PFD))	31
6.3. Projektqualitätsplan	31
6.3.1. Qualitätsverantwortlichkeiten	31
6.3.2. Standards	31
6.3.3. Schlüssel Qualitätskriterien	31
6.3.4. Qualitätskontrolle	32
6.3.5. Abnahme Kriterien	32
7. Kommunikationsplan	34
8. Risiko- und Notfallplan	35
III. Produktstrukturplan	36
IV. Produktflussdiagramm	39
V. Produktbeschreibung eines Hauptprodukts	42
1. Produkt 1.2.1.: Backend (Framework)	44
1.1. Zweck	44
1.2. Funktionen im einzelnen	44
1.2.1. Utility Klassen	44
1.2.2. Globales singleton Objekt	47
1.3. Ordner-Struktur des Backends	47
1.4. Datenbank-Struktur	48
1.4.1. Tabellen	48
1.4.2. Tabellen-Eigenschaften	50
1.5. Mapping: Datenbank-Tabellen zu DB Beans	50
1.6. Controller und PluginHandler	51
1.7. Plugins	51
1.7.1. Aufbau und Funktionsweise	51
1.7.2. Zusammenspiel von Plugin und Layout	53

VI. Projektplan als Gantt-Chart	54
VII. Risiko-Log	61
1.8. Legende zum Risiko-Log	64
1.9. Notfallplan	64
VIII Issue-Log	65
Abkürzungsverzeichnis	71

Teil I.

Project Brief

Project Brief

PS Framework

Projekt-Nr.:	2005-09
Dok. ID:	001.00
Produkt-Typ:	Management
Version:	0.2
Klassifizierung:	frei zugänglich
Verteiler:	n/a

Autor(en):	Nicolas Lanquetin, HdM
Verantwortliche(r):	Nicolas Lanquetin, HdM
Kunde:	Unipaz Europe

1. Zweck und Geltungsbereich

Der Project Brief liefert eine vollständige und sichere Grundlage für die Initialisierung des Projekts. Des Weiteren bildet er die Grundlage für die Entscheidung der Projektverantwortlichen (Project Board) zur Bevollmächtigung der Erstellung des Project Initiation Documents (PID).

Der Inhalt des Project Briefs wird in das Project Initiation Document übernommen, welches das Arbeitsdokument für das Verwalten und Leiten des Projekts ist, und wird dort erweitert und verfeinert (Siehe Teil II auf Seite 18).

Der Project Brief wird auch in der Initialisierungstage zur Erstellung des Project Quality Plans verwendet.

2. Hintergrund

Viele der privaten Kunden benötigen im Moment einen Internetauftritt. Die Erfahrung hat gezeigt, dass sich für diese Kundenklasse ein CMS am besten eignet. Dieses können sie benutzen, um ihre Inhalte unabhängig von Entwickler und Webmasters zu pflegen.

Das Projekt *PS Framework* ist ein solches CMS. Es ist aus der Anforderung der Internationalisierung entstanden und wird zukünftig eine ältere Anwendung der Organisation Unipaz-Europe ersetzen. Zwei weitere potentielle Privatkunden sind bereits am Produkt interessiert, sodass eine umfangreiche und leicht erweiterbare Applikation benötigt wird.

Das Projekt hat neben ihrem praktischen Nutzen ebenfalls das Ziel, die Möglichkeiten des PHP5 zu erforschen und evaluieren. Diese sehr aktuelle Version von PHP ist aufgrund ihrer Unterstützung für Objektorientierung zukunftsweisend und soll daher im frühen Stadium bereits beherrscht werden.

3. Projektdefinition

3.1. Projektziele

Das Projekt hat folgende Ziele:

- Es soll ein CMS aufgesetzt werden mit festgelegten Anforderung:
 - Das Design soll mittels einer Template Engine ohne grossen Zeitaufwand geändert werden können.
 - Das CMS muss eine intuitive und benutzerfreundliche Administrationsoberfläche haben. Diese ermöglicht
 - * das Editieren der Inhalte,
 - * das Erstellen von Administratorbenutzern und das Zuweisen selbstbestimmbarer Benutzerrollen und -rechte,
 - * das Managen von Medien (Bilder, Animationen, Office-Dokumente, Audio-Dateien, usw.)
 - Es muss ein Rollen- und Rechtesystem geben, bei welchem Benutzer und Gruppen erstellt und die Management- und Administrations-Rechte gezielt vergeben werden können.
- Die Möglichkeiten von PHP Version 5 bis 5.1 sollen evaluiert und effektiv eingesetzt werden.

3.2. Projektumfang

Das Projekt wird etwa ein Jahr in Anspruch nehmen. Es wird jedoch im Vergleich zu bestehenden Content Management Systemen weniger umfangreich sein. Da keine dringenden Nachfragen der Kunden bestehen, hat die Umsetzung des Projekts wesentlich mehr Zeit und rückt in der Priorität daher stark nach hinten.

3.3. Project Approach

Siehe Tabelle 3.1 für eine Auflistung der wichtigsten Aspekte für die Projektansatz Entscheidung.

3. Projektdefinition

Beeinflussende Aspekte	Erklärung und Erläuterungen zu Auswirkung und Einfluss
Folgerungen / Notwendigkeiten fürs Business	Durch das Projekt können weitere Kunden gewonnen werden
Kundenstrategie (kurz- und langfristig)	Langfristige Kundenstrategie durch modularer Erweiterbarkeit des Projekts
Vorhandene Infrastruktur & Betrieb	LAMP-Server (gemietet, oder selbst gehostet & gewartet)
Einschränkung der Zeit	Realisierungs des Basisgerüsts bis Anfang 2006
Einschränkungen der Mittel	Keine Einschränkung
Einschränkungen der Qualität / Anpassungsfähigkeit	Keine Qualitätseinschränkungen
Einschränkung der Ressourcen Nutzung oder Verfügbarkeit	Produkt ist für kleine Kunden zugeschnitten, wo Ressourcen nur sehr wenig verbraucht werden. Die Verfügbarkeit muss jedoch hoch sein.
Risiko Erwägungen	Siehe Risiko-Log (Teil VII Seite 62)
Sicherheitsanforderungen	Authentifizierung für das Management und die Administration
Industrie / Kunden / Standards	Zielgruppe: Privatkunden. Eigene Standards werden angewandt.
Externe Abhängigkeiten	Entwickler von der FH Reutlingen & externes -Editor und Media-Management Tool
Wartungs- & Betriebsanforderungen	Wartung wird generell nicht benötigt. Betriebsanforderungen richten sich nach Kunden
Schnittstellen mit anderen Projekten / Tätigkeiten	Keine Schnittstellen
Anforderung an Qualifikation und Ausbildung	Kunde muss mit dem Umgang eines Webbrowser und Content Management Systemen vertraut sein.
Einführung und Übergabe	Nach Absprache. Generell: Applikation wird auf gemieteten LAMP-Server installiert und lauffähig gemacht
Erwartete Dauer des Projekts	1 Jahr

Tabelle 3.1.: Auflistung der wichtigsten Aspekte für die Projektansatz Entscheidung

Gewählter Project Approach

Es wird ein Produkt-Approach gewählt. Jeder Projektteil ist ein Produkt. Das Zusammenspiel aller Teilprodukte führt zum Endprodukt. Im weiteren Verlauf des Projekts wird darum ein Produktstrukturplan und Produktflussdiagramm erstellt werden.

Die Wahl des Produkt-Approach hat den Vorteil, dass alle Teilprodukte klar definiert und leicht überprüfbar sind.

3.4. Ausschüsse

- Es wird keine Anbindung an bereits bestehende Authentifizierungssysteme unterstützt (z.B. LDAP)
- Vom Kundem selbst gehostete LAMP-Systeme werden weder konfiguriert noch gewartet. Insbesondere hat der Kunde fehlende PEAR-Bibliotheken oder PHP-Extensions selbst zu installieren und ihre Lauffähigkeit zu gewährleisten.

3.5. Einschränkungen

- Die Applikation kann nur auf *einem* Server installiert werden (Kein Load-Balancing möglich).

3.6. Voraussetzungen

- Zur Installierung der Applikation ist ein LAMP-Server nötig. Dieser muss vom Kunde entweder gemietet, oder vom Kunde selbst eingerichtet und gewartet werden.
- Die Benutzer der Anwendung müssen mit dem Umgang eines Webbrowser und Content Management Systemen vertraut sein.

4. PS Business Case

Das Produkt soll für die Friedensorganisation Unipaz-Europe kostenlos erstellt werden. Des Weiteren gibt es bereits 2 weitere potenzielle Kunden, die an diesem Produkt interessiert wären.

Tabelle 5.1 zeigt den nicht-finanziellen Nutzen und wie er gemessen wird.

Nutzen:	Messung des Nutzens:
Praktische Erfahrung mit aktueller Technologie (PHP Version 5)	Eingesparte Zeit für zukünftige PHP-Projekte
Referenz/Portfolio	Gewinnung von neuen Kunden

Tabelle 4.1.: PS Business Case - Nicht-finanzieller Nutzen

5. Qualitätserwartungen

Der Kunde hat folgende Qualitätserwartungen, welche vom Projekt erfüllt werden müssen:

- *Usability*
Die Applikation muss benutzerfreundlich und intuitiv bedienbar sein. Für Administratoren muss der rudimentäre Umgang mit einem CMS geläufig sein. Ihnen sollte jedoch eine Hilfe zur Verfügung stehen, welche alle möglichen Aktionen erläutert.
- *Zeitbegrenzungen*
Die Aufrufe der durch die Applikation generierten Webseiten müssen schnell sein.
- *Sicherheit*
Die Applikation muss Angriffssicher sein.
- *Skalierbarkeit*
Das Hinzufügen neuer Features (z.B. Gästebuch, Kalender, usw) soll reibungslos und einfach zu realisieren sein.
- *Zuverlässigkeit und Verfügbarkeit*
Die Applikation sollte in keinem Fall aufgrund des Produkts selbst ausfallen. Ausnahme stellt nur ein durch uns unabhängiges Ausfallen des Servers, ein Upgrade des Servers oder ein Update der Applikation dar.
- *Kapazität*
Die Applikation soll so performant sein, dass mehrere simultane Zugriffe ohne teurer Hardware möglich sind.

6. Abnahmekriterien

Die Abnahme des Produkts erfolgt, wenn die Hauptfunktionen implementiert sind und fehlerfrei laufen. (siehe Abschnitt 3.1)

Die erste Abnahme kann bereits ohne Implementierung einer Administrationsoberfläche erfolgen. Es muss jedoch gewährleistet sein, dass die Inhalte über ein externes Datenbank-Tool gepflegt werden können.

Funktionalität: Folgende Funktionalität muss für die Abnahme zwingend umgesetzt worden sein:

- Die Website muss ein Layout konform des Style Guide des Kunden haben. Dabei muss das Kundenlogo eingebunden werden und das Design entsprechend der Logo-Farben umgesetzt werden.
- Ein hierarchisches Menü muss implementiert werden. Die Anzahl der Menü-Elemente darf nicht begrenzt sein.
- Zu jedem Menü-Punkt existiert ein Inhalt.
- Dem Besucher muss es möglich sein zwischen den Sprachen Englisch und Französisch aussuchen zu können.
- Die Implementierung beruht auf festgelegten Standards: Das Layout soll XHTML-valid sein und der PHP-Code muss entsprechend dem PEAR Coding Standard formatiert werden. Besonders hohen Wert gilt es auch auf die Ordnerstruktur zu legen, sodass sie den anderen Projekten ähnelt.

Qualitätskriterien: Zudem müssen folgende Qualitätskriterien gelten:

- Das generieren einer durchschnittlich vollen Seite sollte nicht 150 ms überschreiten.
- Die Applikation soll mindestens 50 simultane Benutzeranfragen gleichzeitig verarbeiten können.
- Besondere Rücksicht muss auf die Sicherheit genommen werden, insbesondere für das Einhacken über URL Parameter und in Bezug auf das Session Hijacking.
- Des Weiteren gelten die in Kapitel 5 genannten Punkte bezüglich der Qualitätserwartungen.

Abnahme-Medium: Das Produkt wird auf einem LAMP Server installiert. Dabei kann es sich um einen vom Kunden gemieteten Server handeln, welcher von uns gewartet wird, oder einem vom Kunden zur Verfügung gestellten Server, auf welchen wir jedoch Zugriffsrechte bekommen. (siehe auch Abschnitt 3.4 zum Thema Ausschüsse)

Umfang: Der Kunde erhält den gesamten Sourcecode des Projekts. Dieser beinhaltet mindestens den Quellcode, und je nach Absprache weitere Dokumente zum Weiterentwickeln des Produkts.

7. Risiken

Eine "Proposal Risk Analysis" (PRA) wurde nicht durchgeführt.

Folgende Risiken wurden erkannt:

- *Technische Risiken*
 - Die benötigten Freeware Tools zum Editieren von XHTML-Validen Text und Managen von Medien kann nicht in kommerziellen Anwendungen eingesetzt werden, oder entspricht nicht den Anforderungen.
- *Externe Risiken*
 - Ein für das Front-End benötigter Mitarbeiter ist nicht verfügbar.
- *Interne Risiken*
 - Durch die Arbeit an anderen Projekten, oder das Vorbereiten auf Prüfungen, kann das Projekt kurzzeitig vernachlässigt werden.

Im Verlauf des Projekts wird ein Risiko-Log erstellt, in welchem später erkannte Risiken noch weiter aufgezeichnet werden.

Teil II.

Project Initiation Document

Project Initiation Document

PS Framework

Projekt-Nr.:	2005-09
Dok. ID:	002_00
Produkt-Typ:	Management
Version:	0.6
Klassifizierung:	frei zugänglich
Verteiler:	n/a

Autor(en):	Nicolas Lanquetin, HdM
Verantwortliche(r):	Nicolas Lanquetin, HdM
Kunde:	Unipaz Europe

1. Zweck und Geltungsbereich

Das Project Initiation Document (PID) definiert das Projekt, legt die Grundlage für dessen und ermöglicht es den gesamten Erfolg einzuschätzen.

Es gibt zwei vorrangige Anwendungen dieses Dokuments:

- Um sicherzustellen, dass das Projekt auf einer sauberen Basis steht, bevor das Project Board angefragt wird, die Hauptverantwortung für das Projekt zu übernehmen
- Als ein Grundlagendokument für das Project Board und um dem Project Manager zu helfen den Fortschritt einzuschätzen, Managementbelange zu ändern und laufende Durchführbarkeitsfragen zu werten.

2. Hintergrund

Viele der privaten Kunden benötigen im Moment einen Internetauftritt. Die Erfahrung hat gezeigt, dass sich für diese Kundenklasse ein CMS am besten eignet. Dieses können sie benutzen, um ihre Inhalte unabhängig von Entwickler und Webmasters zu pflegen.

Das Projekt *PS Framework* ist ein solches CMS. Es ist aus der Anforderung der Internationalisierung entstanden und wird zukünftig eine ältere Anwendung der Organisation Unipaz-Europe ersetzen. Zwei weitere potentielle Privatkunden sind bereits am Produkt interessiert, sodass eine umfangreiche und leicht erweiterbare Applikation benötigt wird.

Das Projekt hat neben ihrem praktischen Nutzen ebenfalls das Ziel, die Möglichkeiten des PHP5 zu erforschen und evaluieren. Diese sehr aktuelle Version von PHP ist aufgrund ihrer Unterstützung für Objektorientierung zukunftsweisend und soll daher im frühen Stadium bereits beherrscht werden.

3. Projektdefinition

3.1. Projektziele

Das Projekt hat folgende Ziele:

- Es soll ein CMS aufgesetzt werden mit festgelegten Anforderung:
 - Das Design soll mittels einer Template Engine ohne grossen Zeitaufwand geändert werden können.
 - Das CMS muss eine intuitive und benutzerfreundliche Administrationsoberfläche haben. Diese ermöglicht
 - * das Editieren der Inhalte,
 - * das Erstellen von Administratorbenutzern und das Zuweisen selbstbestimmbarer Benutzerrollen und -rechte,
 - * das Managen von Medien (Bilder, Animationen, Office-Dokumente, Audio-Dateien, usw.)
 - Es muss ein Rollen- und Rechtesystem geben, bei welchem Benutzer und Gruppen erstellt und die Management- und Administrations-Rechte gezielt vergeben werden können.
- Die Möglichkeiten von PHP Version 5 bis 5.1 sollen evaluiert und effektiv eingesetzt werden.

3.2. Projektumfang

Das Projekt wird etwa ein Jahr in Anspruch nehmen. Es wird jedoch im Vergleich zu bestehenden Content Management Systemen weniger umfangreich sein. Da keine dringenden Nachfragen der Kunden bestehen, hat die Umsetzung des Projekts wesentlich mehr Zeit und rückt in der Priorität daher stark nach hinten.

3.3. Project Approach

Siehe Tabelle 3.1 für eine Auflistung der wichtigsten Aspekte für die Projektansatz Entscheidung.

3. Projektdefinition

Beeinflussende Aspekte	Erklärung und Erläuterungen zur Auswirkung und Einfluss
Folgerungen / Notwendigkeiten fürs Business	Durch das Projekt können weitere Kunden gewonnen werden
Kundenstrategie (kurz- und langfristig)	Langfristige Kundenstrategie durch modularer Erweiterbarkeit des Projekts
Vorhandene Infrastruktur & Betrieb	LAMP-Server (gemietet, oder selbst gehostet & gewartet)
Einschränkung der Zeit	Realisierungs des Basisgerüsts bis Anfang 2006
Einschränkungen der Mittel	Keine Einschränkung
Einschränkungen der Qualität / Anpassungsfähigkeit	Keine Qualitätseinschränkungen
Einschränkung der Ressourcen Nutzung oder Verfügbarkeit	Produkt ist für kleine Kunden zugeschnitten, wo Ressourcen nur sehr wenig verbraucht werden. Die Verfügbarkeit muss jedoch hoch sein.
Risiko Erwägungen	Siehe Risiko-Log (Teil VII Seite 62)
Sicherheitsanforderungen	Authentifizierung für das Management und die Administration
Industrie / Kunden / Standards	Zielgruppe: Privatkunden. Eigene Standards werden angewandt.
Externe Abhängigkeiten	Entwickler von der FH Reutlingen & externer WYSIWYG-Editor und Media-Management Tool
Wartungs- & Betriebsanforderungen	Wartung wird generell nicht benötigt. Betriebsanforderungen richten sich nach Kunden
Schnittstellen mit anderen Projekten / Tätigkeiten	Keine Schnittstellen
Anforderung an Qualifikation und Ausbildung	Kunde muss mit dem Umgang eines Webbrowser und Content Management Systemen vertraut sein.
Einführung und Übergabe	Nach Absprache. Generell: Applikation wird auf gemieteten LAMP-Server installiert und lauffähig gemacht
Erwartete Dauer des Projekts	1 Jahr

Tabelle 3.1.: Auflistung der wichtigsten Aspekte für die Projektansatz Entscheidung

Gewählter Project Approach

Es wird ein Produkt-Approach gewählt. Jeder Projektteil ist ein Produkt. Das Zusammenspiel aller Teilprodukte führt zum Endprodukt. Im weiteren Verlauf des Projekts wird darum ein Produktstrukturplan und Produktflussdiagramm erstellt werden.

Die Wahl des Produkt-Approach hat den Vorteil, dass alle Teilprodukte klar definiert und leicht überprüfbar sind.

3.4. Projekt Lieferergebnisse

- Unipaz-Europe Organisation
 - Internationalisierung der Organisation
 - Erweiterung der Interessenten
 - Schnelle, kostengünstigere Aktualisierung der Organisation-Ereignisse und Neuigkeiten

3.5. Ausschlüsse und Einschränkungen

- Es wird keine Anbindung an bereits bestehende Authentifizierungssysteme unterstützt (z.B. LDAP)
- Vom Kundem selbst gehostete LAMP-Systeme werden weder konfiguriert noch gewartet. Insbesondere hat der Kunde fehlende PEAR-Bibliotheken oder PHP-Extensions selbst zu installieren und ihre Lauffähigkeit zu gewährleisten.
- Die Applikation kann nur auf *einem* Server installiert werden (Kein Load-Balancing möglich).

3.6. Voraussetzungen und Annahmen

- Zur Installation der Applikation ist ein LAMP-Server nötig. Dieser muss vom Kunde entweder gemietet, oder vom Kunde selbst eingerichtet und gewartet werden.
- Die Benutzer der Anwendung müssen mit dem Umgang eines Webbrowsers und Content Management Systemen vertraut sein.
- Es muss mindestens einen Zuständigen für die Pflege der CMS-Inhalte geben. In einem ersten Release der Produkts wird noch keine Administrationsoberfläche zur Verfügung stehen. Der Zuständige muss aus diesem Grund dem Umgang mit HTML beherrschen und sich mit der Bedienung eines Datenbank-Tools, sowie dem Datenbank-Aufbau vertraut machen.

3.7. Externe Abhängigkeiten und Schnittstellen

Das Projekt wird einen externen Javascript WYSIWYG Editor benötigen. Dabei kommt der sehr populäre Editor *TinyMCE* in Frage, welcher darüber hinaus noch andere nützliche Funktionalitäten mit sich bringt, wie z.B. die Verwaltung von Medien-Dateien.

4. Projekt Management Team Struktur und Rollen-Definitionen

4.1. Projekt Beteiligte

Projektmitarbeiter

- **Nicolas Lanquetin:** *Student der Medieninformatik an der Hochschule der Medien Stuttgart*
Projektleiter und Programmierer im Backend des Frameworks
- **Jochen Kornau:** *Student der Medien- und Kommunikationsinformatik der FH Reutlingen*
Zuständig für das Frontend des Frameworks (Layout: Public Access, Administrative Access)

Relevante Kundenmitglieder

- **Bernard H.:**
Direktor von Unipaz Europe
- **Petra M.:**
Verantwortlich für das Pflegen der Inhalte
- **Frédéric K.:**
Schatzmeister

Kontaktinformationen der Beteiligten

Name	Telefon	E-Mail
Nicolas Lanquetin Jochen Kornau	0185 1416073 *****	psf@psbase.com *****
Bernard H. Petra M. Frédéric K.	***** ***** *****	*****@unipaz-europe.org *****@unipaz-europe.org *****@unipaz-europe.org

Tabelle 4.1.: Kontaktinformationen der Beteiligten

5. Überblick Business Case

Das Produkt soll für die Friedensorganisation Unipaz-Europe kostenlos erstellt werden. Des Weiteren gibt es bereits 2 weitere potenzielle Kunden, die an diesem Produkt interessiert wären.

Tabelle 5.1 zeigt den nicht-finanziellen Nutzen und wie er gemessen wird.

Nutzen:	Messung des Nutzens:
Praktische Erfahrung mit aktueller Technologie (PHP Version 5)	Eingesparte Zeit für zukünftige PHP-Projekte
Referenz/Portfolio	Gewinnung von neuen Kunden

Tabelle 5.1.: PS Business Case - Nicht-finanzieller Nutzen

Dieser Überblick des Business Case liefert eine Grundlage für Vergleiche mit späteren Überarbeitungen des Business Case.

6. Project Plan

6.1. Planübersicht

6.1.1. Management Zusammenfassung

Folgende Vorgehensweise wurde zur erfolgreichen Realisierung des Projekts gewählt:

In einer ersten Phase müssen sich die Mitarbeiter gründlich mit den eingesetzten Technologien auseinandersetzen. Anschließend wird im Rahmen einer Analyse festgelegt wie die Umsetzung spezifiziert wird.

Erst dann wird mit der Entwicklungsphase begonnen. Zuerst wird der Backend-Bereich realisiert, der alle zum weiteren Vorgehen nötigen Klassen und Bibliotheken zur Verfügung stellt. Darauf aufbauend muss der öffentliche Zugang des CMS realisiert werden. Dieser beinhaltet eine klare Aufteilung der Funktionalitäten in sog. Plugins. Zudem wird in dieser Phase auch am Layout gearbeitet, das durch Templates realisiert wird. Nach ausführlichem Testen, kann dann ein ersten Release stattfinden.

Ist der öffentliche Zugang realisiert, muss der Administrative Zugang implementiert werden. In diesem soll es erstmal möglich sein die Inhalte unabhängig von einem Datenbank-Tool zu verwalten. Ferner werden hier auch Möglichkeiten geboten bestimmte Rechte-Einstellungen vorzunehmen. Ist dieser Abschnitt ebenfalls realisiert, kann eine erste fertige Version geliefert werden.

Nach erfolgreicher Umsetzung der oben beschriebenen Punkte, kann das Projekt je nach Anforderungen erweitert werden.

6.1.2. Planungsvoraussetzungen und -annahmen

Die beteiligten Entwickler haben bereits Erfahrung und Kenntnisse im Umgang mit LAMP-Systemen und den Technologien PHP und PEAR, sowie XHTML und der Smarty Template Engine. Die Aufgaben wurden dementsprechend auf die Entwickler aufgeteilt.

Es besteht von Kundenseite kein dringendes Bedürfnis nach einer Administrationsoberfläche, solange gewährleistet ist, dass die Inhalte von uns gepflegt werden können.

6.1.3. Externe Abhängigkeiten

Vom Erfolg des Projekts abhängig, ist der Beitrag im Layoutbereich eines externen Mitarbeiters. Des Weiteren muss eine starke Zusammenarbeit mit den Inhaltverantwortlichen von Unipaz Europe gewährleistet sein.

6.2. Projekt Resultate / Produkte

Nachfolgende Tabelle nennt die wichtigsten Resultate / Produkte des Projekts.

Projekt Produkte	Kommentare
Schulung/Einarbeitung	
Software	Die Software kann bereits ausgeliefert werden, ohne dass alle Teilprodukte implementiert worden sind.
Dokumentation	
Umgebung	

Tabelle 6.1.: Projekt Resultate / Produkte

6.2.1. Beschreibung der Hauptprodukte

Tabelle 6.2 beschreibt das zu liefernde Endprodukt des Projekts:

Produktname	1. PS Framework 1.0
Zweck	Endprodukt welches als CMS eingesetzt werden kann
Zusammensetzung	Das Produkt ist zusammengesetzt aus Schulung/Einarbeitung, Software, Dokumentation und Umgebung
Format/Präsentation	Webanwendung welche gemeinschaftliche Erstellung und Bearbeitung von Text- und anderen Multimedia-Dokumenten ("Content") ermöglicht und organisiert
Zugeweiht zu	Nicolas Lanquetin
Qualitätskriterien	Siehe Abschnitt 6.3
Erforderliche Qualitätsprüfungen	Finale Tests auf Ziel-System

Tabelle 6.2.: Produktbeschreibung - PS Framework 1.0

Die Tabellen 6.3, 6.4, 6.5 und 6.6 beschreiben die wichtigsten Teilprodukte, die in diesem Projekt erstellt werden.

Produktname	1.1. Schulung und Einarbeitung
Zweck	Einarbeitung in etablierte Technologien (PHP, Smarty, PEAR, XHTML)
Zusammensetzung	
Format/Präsentation	
Zugewiesen zu	Nicolas Lanquetin, Jochen Kornau
Qualitätskriterien	Verständnis der Technologien und neue Ideen
Erforderliche Qualitätsprüfungen	Die Technologien und Standards müssen verstanden worden sein. Dies muss regelmäßig überprüft und verfolgt werden.

Tabelle 6.3.: Produktbeschreibung - Schulung und Einarbeitung

Produktname	1.2. Software
Zweck	Stellt das Kernstück des Projekts dar. Die Software ist das eigentliche CMS was der Kunde graphisch zu sehen bekommt.
Zusammensetzung	Die Software besteht aus dem Backend (dem Framework), sowie einen öffentlichen und administrativen Zugang
Format/Präsentation	Die Software wird in Form von Sourcecode dem Kunden zur Verfügung gestellt und auf einem Server installiert, sodass der Code interpretiert werden kann und ein visuelles Ergebnis liefert.
Zugewiesen zu	Nicolas Lanquetin, Jochen Kornau
Qualitätskriterien	Geschwindigkeit, Sicherheit, Zuverlässigkeit, Verfügbarkeit, Skalierbarkeit, Kapazität
Erforderliche Qualitätsprüfungen	regelmäßiges Testen der Software anhand von Unit-Tests (PHPUnit) und Lasten-Tests.

Tabelle 6.4.: Produktbeschreibung - Software

Produktname	1.3. Dokumentation
Zweck	Festhalten des Projektverlaufs durch konsequente Aktualisierung der Projekt Management Dokumentation und besseren Einstieg von neuen Programmieren durch das Developer Manual.
Zusammensetzung	Die Dokumentation setzt sich zusammen aus den Projektmanagement-Dokumenten und dem Developer Manual
Format/Präsentation	Die Dokumente werden mit \LaTeX konform des Styleguide realisiert und als pdf zur Verfügung gestellt.
Zugeteilt zu	Nicolas Lanquetin
Qualitätskriterien	Aktualität und Korrektheit der Informationen
Erforderliche Qualitätsprüfungen	regelmäßiges prüfen der Informationen

Tabelle 6.5.: Produktbeschreibung - Dokumentation

Produktname	1.4. Umgebung
Zweck	Es wird eine Laufzeitumgebung gebraucht, damit das Projekt lauffähig ist. Jeder Programmierer hat seine eigene Laufzeitumgebung. Des weiteren hat jeder Programmierer vorgegebene Entwicklungsumgebungen und Kommunikationsprogramme zu nutzen.
Zusammensetzung	Zu den Umgebungen gehören Laufzeitumgebung, Entwicklungsumgebung und ebenfalls Programme für die Kommunikation zwischen den Arbeitern.
Format/Präsentation	
Zugeteilt zu	Nicolas Lanquetin, Jochen Kornau
Qualitätskriterien	Standardisierte Software und stabile und vor allem einheitliche Versionen der Programme
Erforderliche Qualitätsprüfungen	

Tabelle 6.6.: Produktbeschreibung - Umgebung

6.2.2. Projekt Gantt Plan (Product Flow Diagram (PFD))

Der Gantt Chart für den Projektverlauf und das Produkt Fluss Diagramm (PFD) werden in jeweils eigenen Dokumenten behandelt. Siehe dazu Dokument Teil VI und Dokument Teil IV.

6.3. Projektqualitätsplan

6.3.1. Qualitätsverantwortlichkeiten

Die Verantwortung, dass die versprochene Qualität eingehalten wird, übernimmt der Leiter des Projekts Nicolas Lanquetin. Dieser Entschluss ist darauf zurückzuführen, dass der Kunde keine hohen Anforderungen an die Qualität hat, das entstehende Projekt jedoch noch für weitere Kunden von Interesse sein wird.

6.3.2. Standards

Folgende Standards müssen eingehalten werden:

Frontend-Bereich: XHTML 1.0 Strict / CSS 2, Smarty Template Engine

Backend-Bereich: Programmieren nach dem PEAR Coding Standard

6.3.3. Schlüssel Qualitätskriterien

Folgende Schlüssel Qualitätskriterien müssen in folgenden Schlüsselprodukten erfüllt sein:

- **Produkt 1.1.: Schulung und Einarbeitung**
 - Erlangte Kenntnisse
 - Neue Ideen
 - Bereitschaft neue Programmier-Features auszuprobieren
- **Produkt 1.2.: Software**
 - Geschwindigkeit
 - Sicherheit
 - Zuverlässigkeit
 - Verfügbarkeit

- Skalierbarkeit
- Kapazität
- **Produkt 1.3.: Dokumentation**
 - Aktualität
 - Exaktheit und Korrektheit
- **Produkt 1.4.: Umgebung**
 - Standardisierte Software
 - Stabile und vor allem einheitliche Versionen der Programme

6.3.4. Qualitätskontrolle

Die Qualität der Produkte wird vor allem durch regelmäßige Tests an der Applikation gewährleistet.

Die Tests beruhen auf mehreren Punkte:

- **Logging**

Anhand der Logs können Fehler schneller gefunden und behoben werden. Zudem kann die durchschnittliche Zeit gemessen werden die für das Ausführen einer Anfrage gebraucht wird.
- **Lasten-Tests**

Die Lastentests können z.B. mit dem frei erhältlichen Programm *JMeter* ausgeführt werden. Hier kann ein realistisches Besucher-Szenario aufgestellt werden, bei dem sich mehrere Benutzer gleichzeitig auf der Webapplikation aufhalten.
- **Unit-Tests**

Anhand der PEAR-Bibliothek *PHPUnit* können gezielte Tests der einzelnen Komponenten gemacht werden.

6.3.5. Abnahme Kriterien

Die Abnahme Kriterien definieren in messbaren Bedingungen, was am Endprodukt erledigt sein muss, so dass es für den Kunden und die davon betroffene Belegschaft annehmbar bzw. akzeptabel ist.

Die Abnahme des Produkts erfolgt, wenn die Hauptfunktionen (siehe Abschnitt 6.2) implementiert sind und fehlerfrei laufen.

Die erste Abnahme kann bereits ohne Implementierung einer Administrationsoberfläche erfolgen. Es muss jedoch gewährleistet sein, dass die Inhalte über ein externes Datenbank-Tool gepflegt werden können.

Funktionalität: Folgende Funktionalität muss für die Abnahme zwingend umgesetzt worden sein:

- Die Website muss ein Layout konform des Style Guide des Kunden haben. Dabei muss das Kundenlogo eingebunden werden und das Design entsprechend der Logo-Farben umgesetzt werden.
- Ein hierarchisches Menü muss implementiert werden. Die Anzahl der Menü-Elemente darf nicht begrenzt sein.
- Zu jedem Menü-Punkt existiert ein Inhalt.
- Dem Besucher muss es möglich sein zwischen den Sprachen Englisch und Französisch aussuchen zu können.
- Die Implementierung beruht auf festgelegten Standards: Das Layout soll XHTML-valid sein und der PHP-Code muss entsprechend dem PEAR Coding Standard formatiert werden. Besonders hohen Wert gilt es auch auf die Ordnerstruktur zu legen, sodass sie den anderen Projekten ähnelt.

Qualitätskriterien: Zudem müssen folgende Qualitätskriterien gelten:

- Das generieren einer durchschnittlich vollen Seite sollte nicht 150 ms überschreiten.
- Die Applikation soll mindestens 50 simultane Benutzeranfragen gleichzeitig verarbeiten können.
- Besondere Rücksicht muss auf die Sicherheit genommen werden, insbesondere für Das Einhacken über URL Parameter und in Bezug auf das Session Hijacking.
- Des Weiteren gelten die in Kapitel 5 genannten Punkte bezüglich der Qualitätserwartungen.

Abnahme-Medium: Das Produkt wird auf einem LAMP Server installiert. Dabei kann es sich um einen vom Kunden gemieteten Server handeln, welcher von uns gewartet wird, oder einem vom Kunden zur Verfügung gestellten Server, auf welchen wir jedoch Zugriffsrechte bekommen. (siehe auch Abschnitt 3.4 zum Thema Ausschüsse)

Umfang: Der Kunde erhält den gesamten Sourcecode des Projekts. Dieser beinhaltet mindestens den Quellcode, und je nach Absprache weitere Dokumente zum weiterentwickeln des Produkts.

7. Kommunikationsplan

Folgender Kommunikationsplan bestimmt alle Parteien, für die das Projekt von Bedeutung ist, sowie die Mittel und Häufigkeit der Kommunikation zwischen ihnen und dem Projekt. Der Kommunikationsplan bestimmt die externe Kommunikation.

Tabelle 7.1 zeigt den Kommunikationsplan:

Quelle	Empfänger	Inhalt	Häufigkeit	Methode
Petra M.	Nicolas L.	CMS-Inhalte	nach Bedarf	Email
Nicolas L.	Jochen K.	Überprüfen des Stands	wöchentlich	Skype
Nicolas L.	Unipaz Europe	Informationen zu neuen Releases und Fortschritt	monatlich	Email

Tabelle 7.1.: Kommunikationsplan

8. Risiko- und Notfallplan

Die Risiken und Notfallpläne werden in einem eigenen Dokument festgehalten. Dieses ist im Dokument Teil [VII](#) zu finden.

Teil III.

Produktstrukturplan

Produktstrukturplan (PSP)

PS Framework

Projekt-Nr.: 2005-09
Dok. ID: 003_00
Produkt-Typ: Management
Version: 0.4
Klassifizierung: frei zugänglich
Verteiler: n/a

Autor(en): Nicolas Lanquetin, HdM
Verantwortliche(r): Nicolas Lanquetin, HdM
Kunde: Unipaz Europe

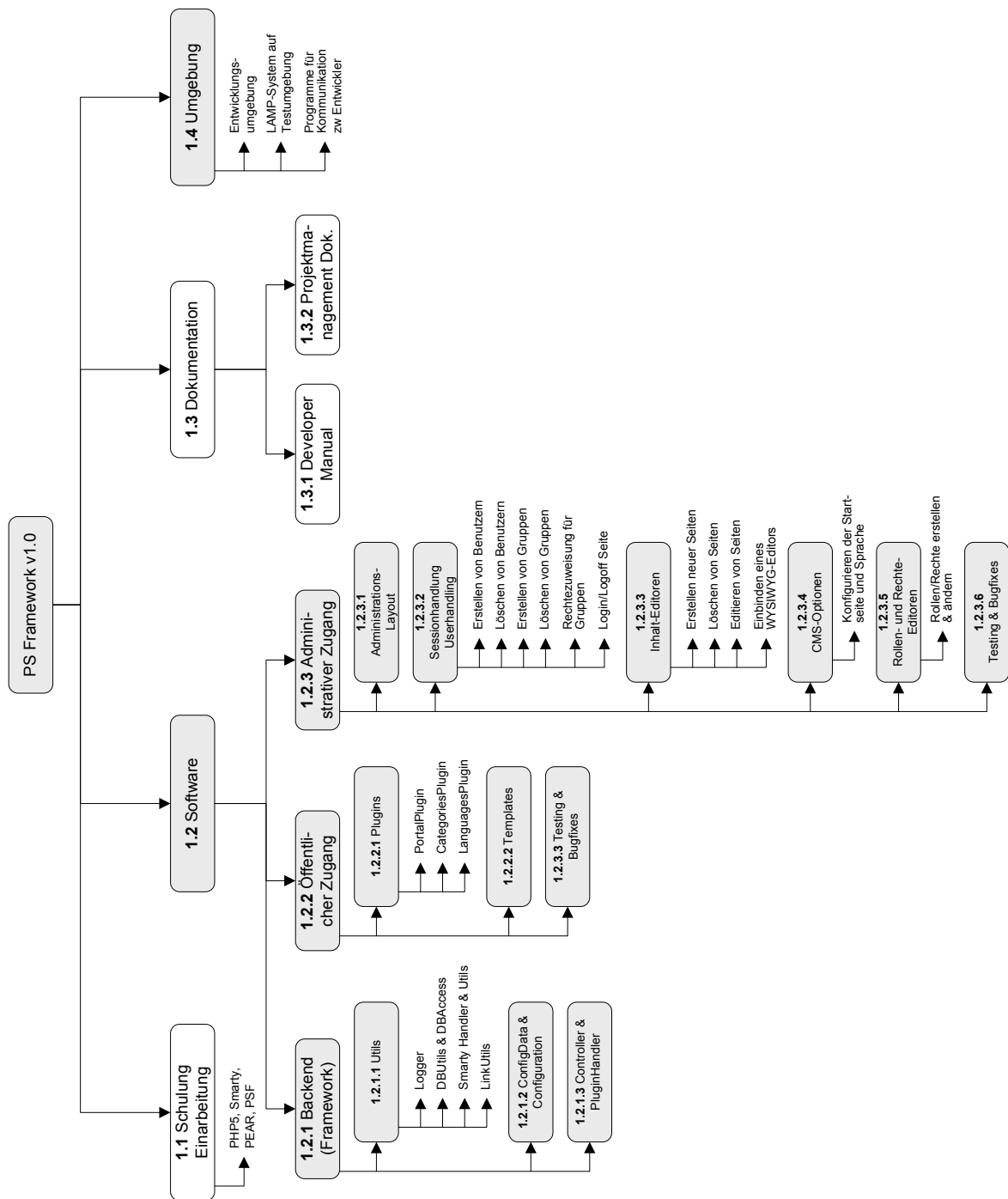


Abbildung 8.1.: Produktstrukturplan

Teil IV.

Produktflussdiagramm

Produktflussdiagramm (PFD)

PS Framework

Projekt-Nr.:	2005-09
Dok. ID:	004.00
Produkt-Typ:	Management
Version:	0.4
Klassifizierung:	frei zugänglich
Verteiler:	n/a
Autor(en):	Nicolas Lanquetin, HdM
Verantwortliche(r):	Nicolas Lanquetin, HdM
Kunde:	Unipaz Europe

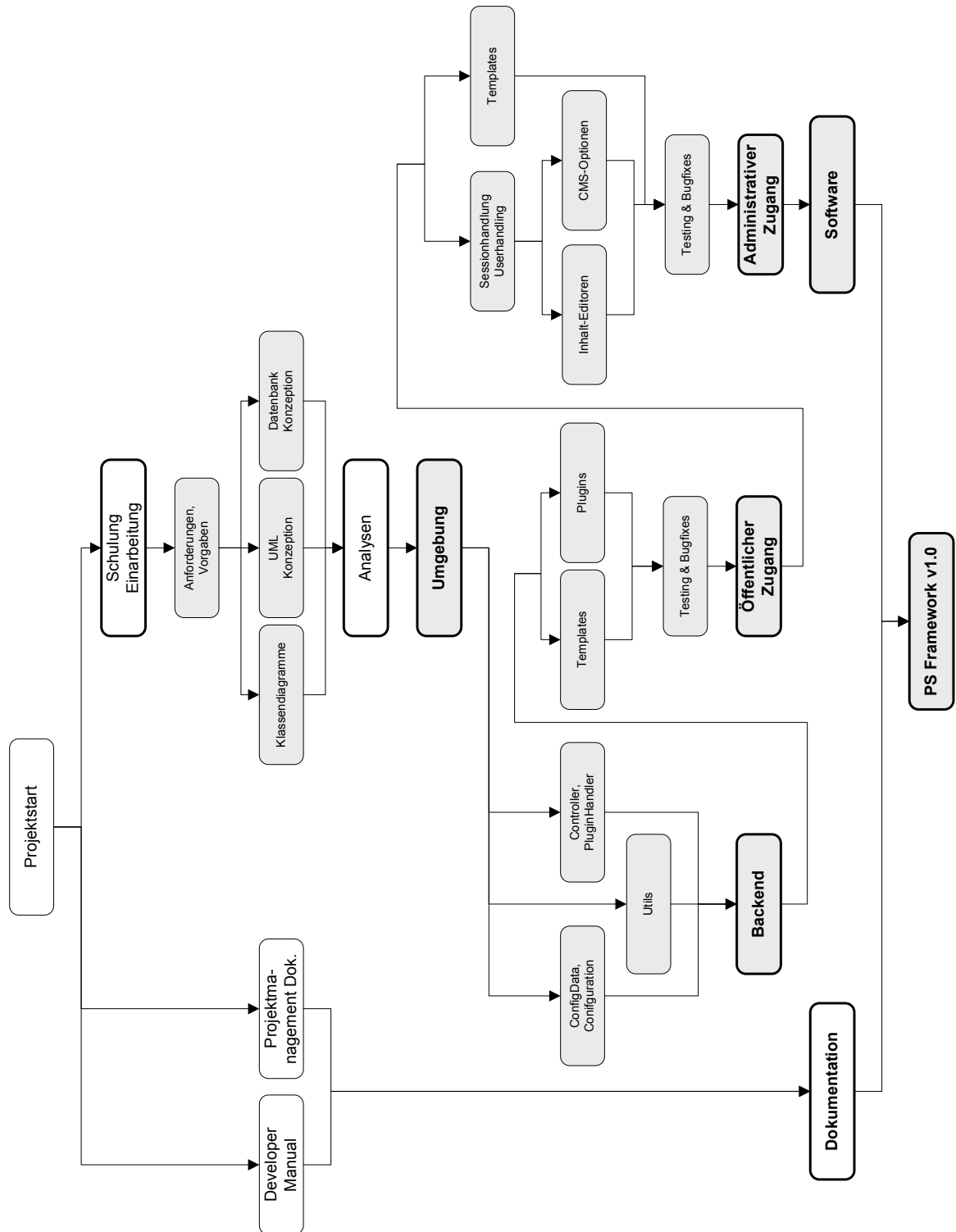


Abbildung 8.2.: Produktflussdiagramm

Teil V.

**Produktbeschreibung eines
Hauptprodukts**

Produkt 1.2.1.: Backend (Framework)

PS Framework

Projekt-Nr.:	2005-09
Dok. ID:	005_00
Produkt-Typ:	Spezifikation / Developer Manual
Version:	0.9
Klassifizierung:	vertraulich
Verteiler:	n/a
Autor(en):	Nicolas Lanquetin, HdM
Verantwortliche(r):	Nicolas Lanquetin, HdM
Kunde:	Unipaz Europe

1. Produkt 1.2.1.: Backend (Framework)

Dieses Kapitel setzt Kenntnisse in der Programmiersprache php und der php Smarty Template Engine voraus. Die Dokumentationen zu diesen Themen sind auf folgenden Seiten einzusehen:

- <http://www.php.net>
- <http://smarty.php.net>

1.1. Zweck

Das Backend, also das eigentliche Framework, liefert alle nötigen Klassen und Bibliotheken, die für die weitere Entwicklung der Applikation notwendig sind. Es ist die Voraussetzung für die Weiterentwicklung am Produkt 1.2.2. (Öffentlicher Zugang) und 1.2.3. (Administrativer Zugang).

1.2. Funktionen im einzelnen

1.2.1. Utility Klassen

ConfigData

Die Klasse `ConfigData` dient der Umwandlung von Konfigurationsdaten aus einer ini-Datei in eine php-Hashmap. Sie muss beim Start eines Aufrufes mit einer ini-Datei initialisiert werden. Anschließend können über Get-Methoden die gewollten Konfigurationsinformationen ausgelesen werden.

Folgende Informationen müssen in der ini-Datei enthalten sein:

Listing 1.1: ini Konfigurationsdatei

```
1 [site_info]
2
3 name = MyCompany
4 publisher = MyCompany
5 copyright = Copyright 2005 MyCompany
6
7 [webserver]
8
9 http_server = http://localhost/mycompany/
```

```
10 https_server = https://localhost/mycompany/
11
12 [filesystem]
13
14 document_root = /path/to/www/
15 libs = /path/to/www/libs/
16 sql = /path/to/sql/
17
18 [database]
19
20 phptype = mysql
21 username = user
22 password = pass
23 hostspec = localhost
24 database = mydatabase
25
26 [logging]
27
28 ; Logging to a file
29 file_enabled = yes
30 file_logfile = /path/to/logfile.log
31 file_threshold = 5
32
33 ; Logging to an email address
34 mail_enabled = no
35 mail_email_from = Site Logger
36 mail_email_to = webmaster_email_here
37 mail_threshold = 1
38
39 [debugging]
40
41 enable_debug = no
42 enable_clrtpl = no
```

Alle zum Betrieb der Applikation notwendigen Informationen sind in der ini-Datei vorhanden. Die wichtigsten Informationen sind sicherlich die Angaben von Pfaden. Jede weitere Informationen die nicht statisch oder technisch notwendig sind, werden in der Datenbank hinterlegt. (siehe dazu auch Abschnitt [1.4](#))

Logger

Die Klasse `Logger` wird benötigt um die Applikationsabläufe zu protokollieren. Diese können sehr hilfreich beim Finden von Bugs sein und bieten zudem Informationen über die Nutzung der Applikation. Es gibt 5 Log Level um die Logging-Informationen zu filtern:

1. `LOG_LEVEL_DEBUG`
2. `LOG_LEVEL_INFO`
3. `LOG_LEVEL_WARNING`

4. LOG_LEVEL_ERROR

5. LOG_LEVEL_ALERT

Der Logger baut auf dem PEAR Log Package auf. Ist das Package auf dem Server nicht vorhanden, muss es in das lib-Verzeichnis kopiert werden (siehe auch Abschnitt 1.3).

Der Logger muss neben dem Loggen in Dateien auch das Versenden von Emails unterstützen, falls ein bestimmter Log Level erreicht ist.

Folgende Daten müssen in der Konfiguration ini-Datei enthalten sein:

Listing 1.2: ini-Datei: Konfiguration für Logger

```
1 [logging]
2
3 ; Logging to a file
4 file_enabled = yes
5 file_logfile = /path/to/logfile.log
6 file_threshold = 5
7
8 ; Logging to an email address
9 mail_enabled = no
10 mail_email_from = Site Logger
11 mail_email_to = webmaster_email_here
12 mail_threshold = 1
```

DBUtils und DBAccess

DBUtils: Das DBUtils Objekt wird ausschliesslich von der Klasse DBAccess genutzt. DBUtils stellt alle Methoden zur Verfügung um Datensätze aus der Datenbank zu erhalten oder in die Datenbank zu schreiben.

Folgende Operationen müssen möglich sein:

- Insert: Einfügen von Datensätzen
- Update: Aktualisieren von Datensätzen
- Select: Erhalten eines ganzen Datensatzes, mehrerer Datensätze oder nur eines Wertes innerhalb eines Datensatzes

Die DBUtils Klasse benutzt das Pear DB Package um diese Operationen durchzuführen. Ist das Package auf dem Server nicht vorhanden, muss es in das lib-Verzeichnis kopiert werden (siehe auch Abschnitt 1.3).

DBUtils muss neben dem gängigen MySQL auch noch weitere Datenbanken unterstützen. Dies wird jedoch durch das Pear DB Package gewährleistet.

Folgende Daten müssen in der Konfiguration ini-Datei enthalten sein:

Listing 1.3: ini-Datei: Konfiguration für DBUtils

```
1 [database]
2
3 phptype = mysql
4 username = user
5 password = pass
6 hostspec = localhost
7 database = mydatabase
```

DBAccess: Die Klasse `DBAccess` wird genutzt um Objekte persistent zu machen. Sie enthält Methoden, die Objekte als Parameter erwartet, und diese in der Datenbank hinzufügt oder aktualisiert, sowie Methoden, die für ein bestimmten Primary Key das gewünschte Objekt (oder eine Liste von Objekten) von der Datenbank holt. Die Objekte die bei den `DBAccess`-Methoden genutzt werden, heissen *DBBeans* und sind im Abschnitt 1.5 genauer beschrieben.

SmartyUtils

Die Klasse `SmartyUtils` ist nur dazu zuständig entweder die gewünschte Seite auszugeben, oder Debugging-Informationen auszugeben, die dem Frontend-Entwickler von Nutzen sind.

1.2.2. Globales singleton Objekt

Die wichtigsten Klassen sind in einem globalen Objekt hinterlegt, welches nach dem singleton Design Pattern implementiert wird.

Beim Start eines Aufrufs werden von den Utility Klassen `ConfigData`, `Logger` und `DBUtils` Instanzen erstellt. Die Reihenfolge ist dabei wichtig, da für das Instanzieren der Klassen `Logger` und `DBUtils` das `ConfigData` Objekt benötigt wird. Um diese Aufgabe kümmert sich das sog. `Glob` Objekt, welches die 3 Utility-Klassen-Instanzen für die Dauer eines Aufrufes speichert.

Dem `Glob` Objekt muss beim ersten Aufruf die ini-Datei übergeben werden. Jeder weitere Aufruf erfolgt über `Glob::singleton()`, mit welchem eine Instanz des `Glob` Objekt zurückgegeben wird.

1.3. Ordner-Struktur des Backends

```
trunk (Subversion Trunk)
: ... content (CMS-Inhalte von Unipaz-Europe)
: ... design (Layout Entwürfe, Graphiken und Style Guides)
```

```
⋮ ... docs (Developer Manuals)
⋮ ... sql (SQL-Dateien zum Erstellen und Befüllen der Datenbank-Tabellen)
⋮ ... www (Das Root-Verzeichnis des Webservers)
⋮ ⋮ ... cache (Smarty Cache)
⋮ ⋮ ... configs (ini Konfigurationsdateien)
⋮ ⋮ ... includes (enthält den Hauptteil des php-Codes)
⋮ ⋮ ⋮ ... dbbeans (Datenbank Beans)
⋮ ⋮ ⋮ ... plugins (Plugins)
⋮ ⋮ ⋮ ... utils (Utility-Klassen)
⋮ ⋮ ... lang (Sprachen Property Dateien)
⋮ ⋮ ... libs (Bibliotheken: PEAR, Smarty, ...)
⋮ ⋮ ... templates (Template Dateien für das Layout)
⋮ ⋮ ... templates_c (Smarty Template Cache)
⋮ ⋮ ... test (PHPUnit Tests)
```

1.4. Datenbank-Struktur

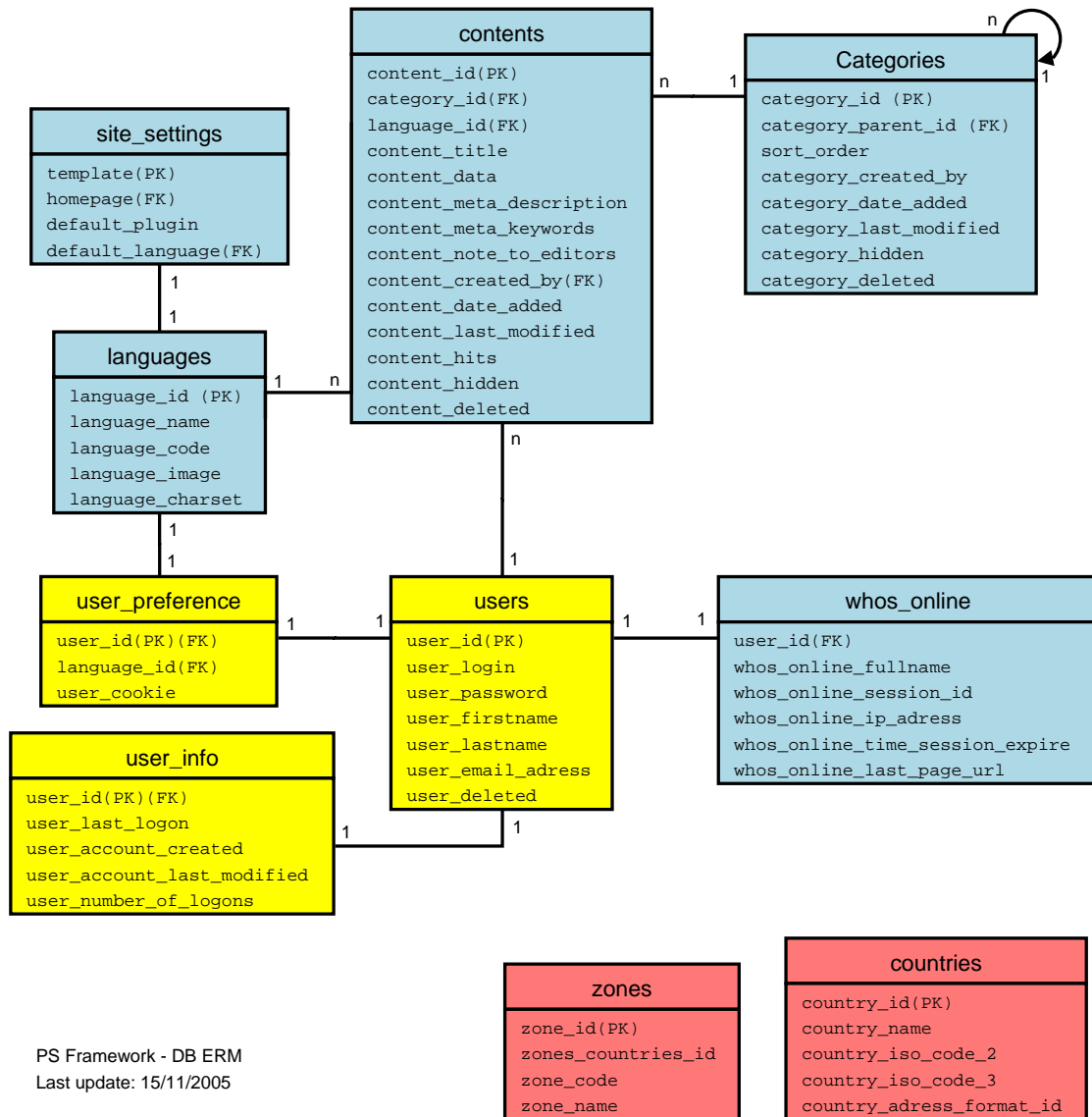
1.4.1. Tabellen

Folgende Tabellen sind für das Backend zwingend notwendig:

- categories
- contents
- languages
- site_settings
- users
- user_info
- user_preference

Das in Abbildung 1.1 gezeigte ERM soll den Aufbau und die Beziehungen zwischen den Tabellen veranschaulichen.

1. Produkt 1.2.1.: Backend (Framework)



PS Framework - DB ERM
Last update: 15/11/2005

Abbildung 1.1.: Database ERM

1.4.2. Tabellen-Eigenschaften

Categories: Das Herzstück der Datenbank-Struktur ist die Categories-Tabelle. Jede Kategorie entspricht einem Punkt im Navigationsmenü des CMS. Es muss bei den Kategorien möglich sein unendlich viele Unterkategorien zu haben. Aus diesem Grund hat jede Kategorie auch eine Referenz auf eine Oberkategorie (Diese darf für Wurzel-Kategorien null sein).

Contents: Zu jeder Kategorie gehört mindestens ein Inhalt aus der Contents-Tabelle (abhängig von der Anzahl Sprachen). Die Contents-Tabelle enthält den Inhalt selbst, sowie Meta-Informationen (keywords, description, comments, editor, date, usw) zu den Inhalten.

Languages: Für jeden Inhalt ist auch eine Sprache festgelegt. Alle unterstützten Sprachen sind in der Language-Tabelle aufgelistet.

In der Tabelle `site_settings` werden Konfigurationen des CMS festgehalten:

- voreingestelltes Plugin
- voreingestelltes Sprache
- die Startseite
- welche Templates für das Layout benutzt werden sollen

Users: Die drei User-Tabellen werden für das spätere Administrieren des CMS benötigt. Sie können bei der Implementierung des Backends allerdings nicht ausgelassen werden, da sie fest mit den Inhalten verankert sind. Dies soll bei der Implementierung von Anfang an berücksichtigt werden.

1.5. Mapping: Datenbank-Tabellen zu DB Beans

Um die Daten der Datenbank-Tabellen nutzen zu können, muss für jede Tabelle eine Klasse geschrieben werden. Diese Klassen werden intern *DBBeans* (Database Beans) genannt und befinden sich im `dbbeans` Ordner (siehe Abschnitt 1.3).

Eine DBBean, spiegelt genau eine Tabelle der Datenbank wieder. Wie bei Beans, besitzen sie neben einem Konstruktor ausschliesslich nur Get- und Set-Methoden. Zudem erben alle DBBeans von der Klasse `AutoGetAndSetBean`, welche den Zugriff auf die Member-Variablen der DBBeans vereinfachen. Die `AutoGetAndSetBean` nutzt dazu die *property and method call overloading* Funktionalität von PHP5.

1.6. Controller und PluginHandler

Der **Controller** ist die Klasse, deren statische Methode `executeRequest()` beim Aufruf einer Seite ausgeführt wird. Der **Controller** initiiert die Smarty Template Engine und ermittelt den Namen des vom Aufrufer gewünschte Plugin. Des Weiteren behandelt es alle nicht abgefangenen Exceptions und gibt diese je nach Einstellung mit Stacktrace im Browser aus.

Für das ausführen eines Plugins wird vom **Controller** die statische Klasse **PluginHandler** genutzt. Diese stellt eine Methode `execute($plugin, $smarty)` zur Verfügung, welche als Parameter den vom **Controller** ermittelten Plugin-Namen und das initialisierte Smarty Objekt erwartet.

1.7. Plugins

Ein Plugin ist eine Klasse welche einfach in das Projekt eingebunden werden kann und eine bestimmte Aufgabe übernimmt.

1.7.1. Aufbau und Funktionsweise

Abbildung 1.2 zeigt den Aufbau des Backends und wie ein Plugin implementiert werden muss.

Es wurde zur Veranschaulichung das Plugin `PortalPlugin` und die Plugin-Komponente `CategoriesPlugin` gewählt. Diese beiden Klassen werden im Produkt 1.2.2. implementiert und gehören nicht zum Backend. Sie erleichtern jedoch das Verständnis des Backends.

Es gibt zwei Typen von Plugins: den eigentlichen **Plugin**, und die **PluginComponents**.

Plugins: Plugins werden von der **PluginBase** abgeleitet. Die **PluginBase** übernimmt fundamentale Funktionen, die bei jedem einzelnen Plugin gebraucht werden. Die **PluginBase** (und somit jedes Plugin) implementiert das **IPlugin** Interface, in welchem die vom **PluginHandler** benötigten Methoden vorgeschrieben sind.

PluginComponents: Jedes Plugin kann zudem noch in Komponenten aufgeteilt werden. Das Plugin `PortalPlugin` hat z.B. eine Komponente `CategoriesPlugin`, welche für das Anzeigen des Navigations-Menü verantwortlich ist. Die Namen der Komponenten können in der Methode `getPluginComponents()` aufgelistet werden und als **String** oder **Array** zurückgegeben werden. Der **PluginHandler** wird die Komponenten dann automatisch laden.

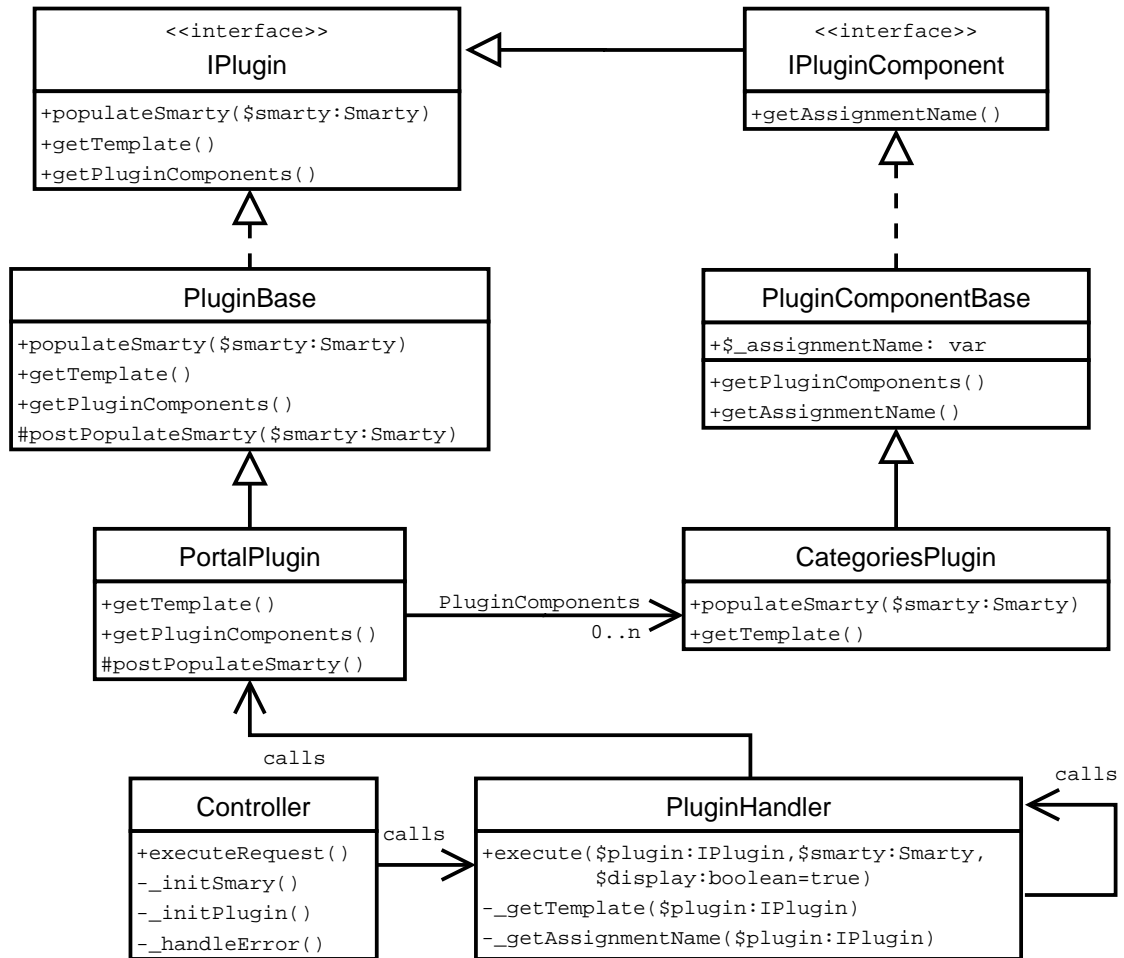


Abbildung 1.2.: PSF UML

1.7.2. Zusammenspiel von Plugin und Layout

Zu jedem Plugin, bzw jeder `PluginComponent`, gehört ein Template. Der Name des Templates muss vom Plugin in der Methode `getTemplate()` zurückgegeben werden. Ein solches Template ist in HTML geschrieben und befindet sich im `templates` Verzeichnis (siehe auch Abschnitt 1.3).

Damit nun dynamische Inhalte innerhalb des Templates angezeigt werden können, muss innerhalb der Methode `populateSmarty($smarty)`, bzw `postPopulateSmarty($smarty)` dem übergeben Smarty-Objekt Werte zugewiesen werden (z.B. Ein Objekt). Diese Werte können dann innerhalb des Templates durch eingebetteten Smarty Code ausgelesen und angezeigt werden.

Teil VI.

Projektplan als Gantt-Chart

Gantt-Chart

PS Framework

Projekt-Nr.:	2005-09
Dok. ID:	006_00
Produkt-Typ:	Management
Version:	0.15
Klassifizierung:	frei zugänglich
Verteiler:	n/a
Autor(en):	Nicolas Lanquetin, HdM
Verantwortliche(r):	Nicolas Lanquetin, HdM
Kunde:	Unipaz Europe

Legende zum Gantt-Chart

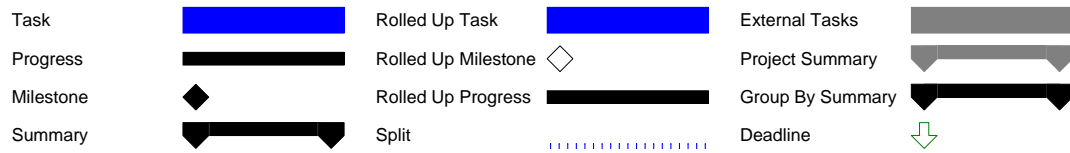


Abbildung 1.3.: Gantt-Chart - Legende

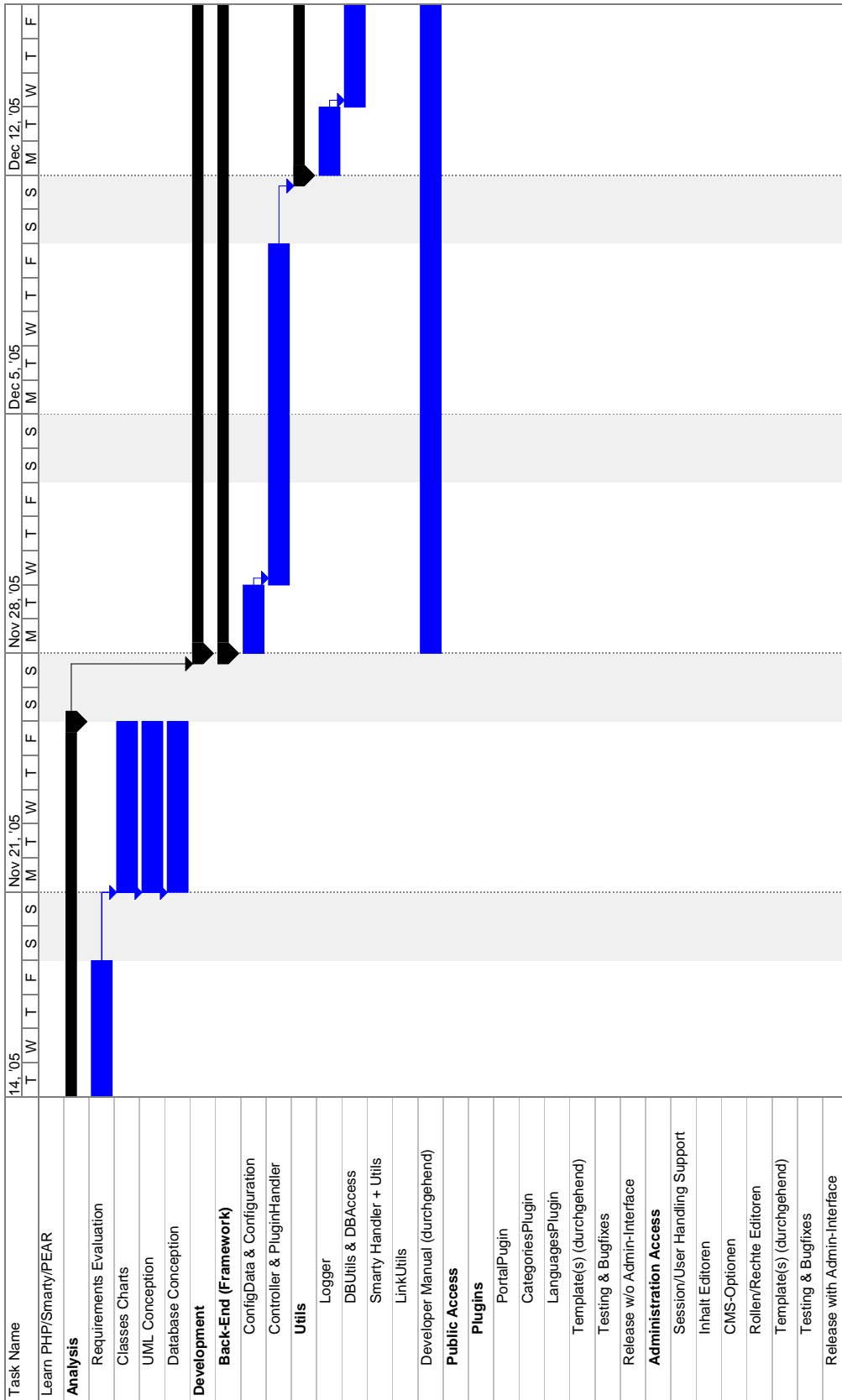


Abbildung 1.5.: Gantt-Chart - Seite 2

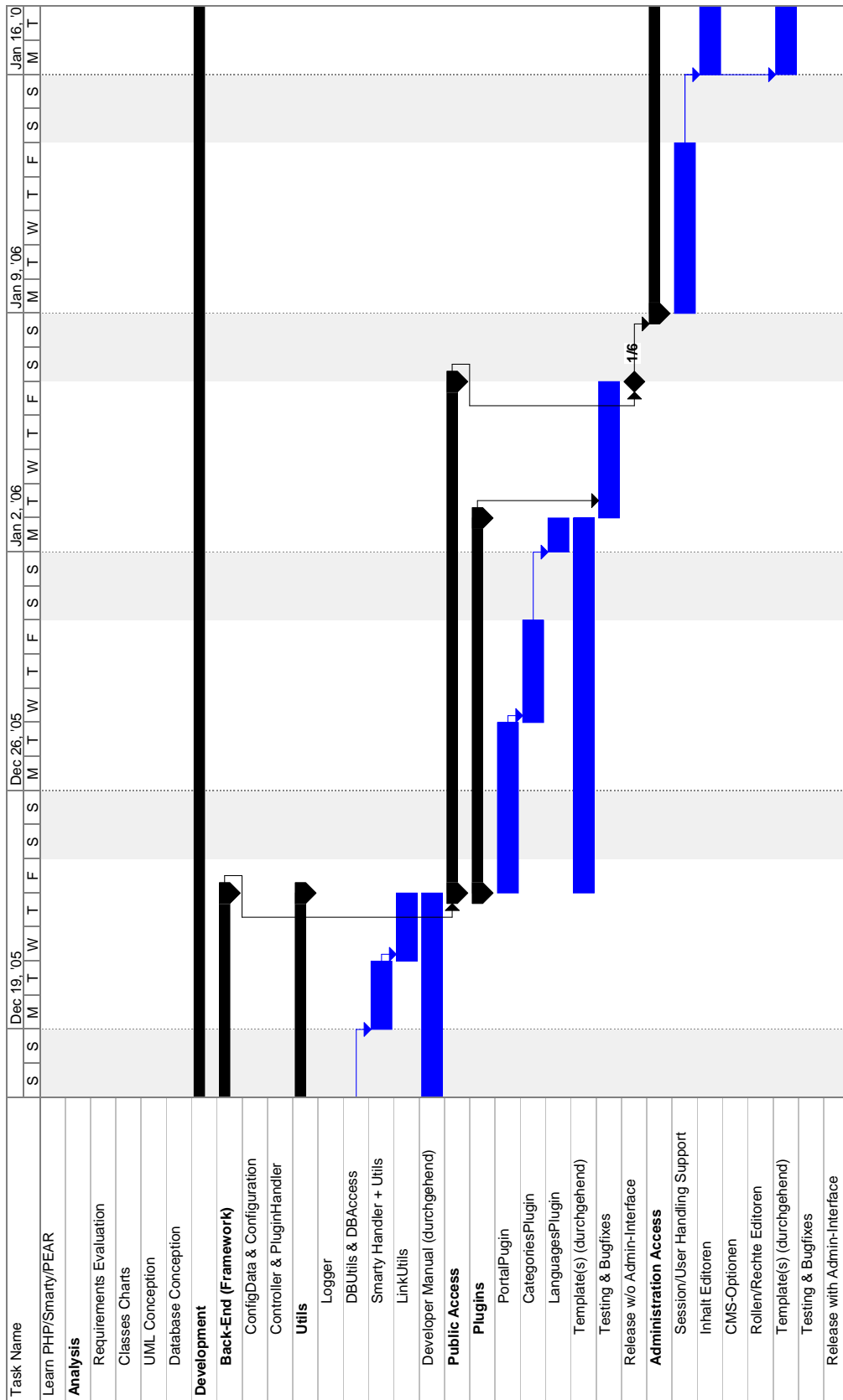


Abbildung 1.6.: Gantt-Chart - Seite 3

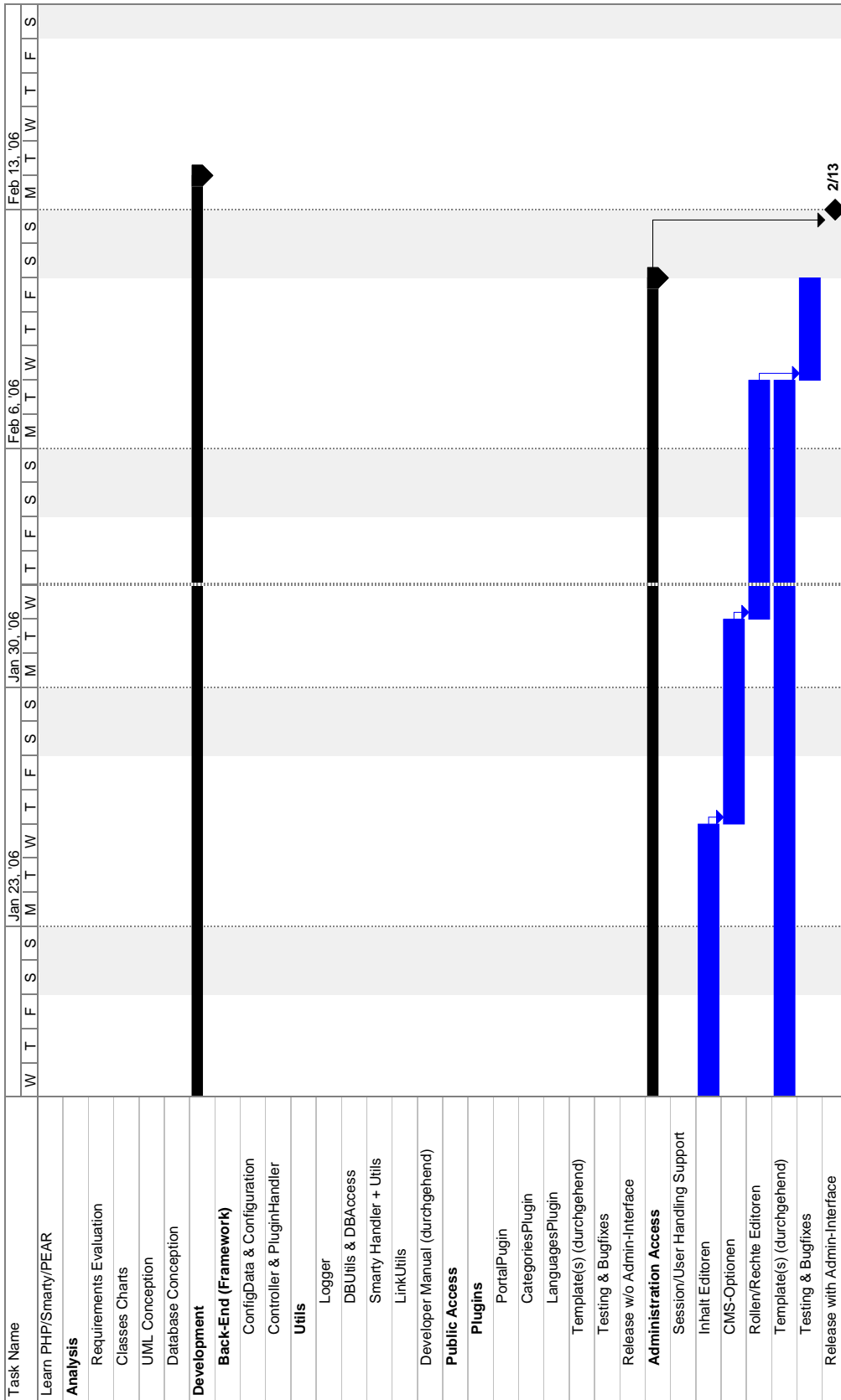


Abbildung 1.7.: Gantt-Chart - Seite 4

Teil VII.
Risiko-Log

Risiko-Log

PS Framework

Projekt-Nr.: 2005-09
Dok. ID: 007_00
Produkt-Typ: Management
Version: 0.4
Klassifizierung: frei zugänglich
Verteiler: n/a

Autor(en): Nicolas Lanquetin, HdM
Verantwortliche(r): Nicolas Lanquetin, HdM
Kunde: Unipaz Europe

Risk Nr.	Autor	Verantwortlicher	Erfasst am	Letzte Änderung	Beschreibung	Wahrscheinlichkeit	Beherrschbarkeit Risikoziffern
T1	Nicolas Lanquetin	Jochen Kornau	7/10/2005		Das WYCIWYG freeware Tool <i>Tiny MCE</i> lässt sich nicht wie gewünscht einbinden	gelb	3
T2	Nicolas Lanquetin	Nicolas Lanquetin	7/10/2005		Session- und Userhandling bringt Komplikationen mit sich	gelb	5
T3	Nicolas Lanquetin	Nicolas Lanquetin	7/10/2005		Einarbeiten in XHTML und der Layer-statt-table-Approach trägt viel Aufwand mit sich	grün	2
T4	Nicolas Lanquetin	Nicolas Lanquetin	7/10/2005		Ungeplante Features werden von dem Framework nicht unterstützt	rot	3
P1	Nicolas Lanquetin	Nicolas Lanquetin	7/10/2005		Die Unterstützung externen Mitarbeiter findet nicht statt	grün	6
E1	Nicolas Lanquetin	Jochen Kornau	7/10/2005		Media-Verwaltung (Bilder, Flash, etc.) lässt sich nicht mit <i>Tiny MCE</i> realisieren	gelb	4

Abbildung 1.8.: Risiko-Log

Abbildung 1.8 zeigt das Risiko-Log.

1.8. Legende zum Risiko-Log

Bewertung der Wahrscheinlichkeit

grün: Das Risiko besteht, die Wahrscheinlichkeit des Eintretens ist gering.

gelb: Wahrscheinlichkeit des Eintretens ist erheblich.

rot: Es besteht eine sehr hohe Wahrscheinlichkeit, dass das negative Ereignis eintritt.

Bewertung der Beherrschbarkeit

1 - Geringes Risiko: Die Auswirkungen sind marginal.

3 - Normales Risiko: Die Auswirkungen sind deutlich spürbar, gefährden jedoch den Projekterfolg nicht.

5 - Erhöhtes Risiko: Die Auswirkungen gefährden den Projekterfolg. Das Risiko ist vom Projektleiter jedoch noch beherrschbar.

7 - Hohes Risiko: Die Auswirkungen gefährden den Projekterfolg. Das Risiko ist vom Projektleiter allein nicht beherrschbar.

1.9. Notfallplan

Für folgende Risiken ist ein Notfallplan und ein Budget vorgesehen falls ein solches Risiko auftreten sollte:

Risk Nr.	Handlung bei Auftritt des Risikos
T4	Die notwendigsten Feature können im Nachhinein noch implementiert werden.
P1	Die Arbeit wird von einem anderem Programmierer übernommen, oder es wird mehr Zeit für die Implementierung eingeplant.
E1	Es muss ein alternatives Tool gefunden werden, oder selber eins programmiert werden.

Tabelle 1.1.: Notfallplan

Teil VIII.
Issue-Log

Issue-Log

PS Framework

Projekt-Nr.:	2005-09
Dok. ID:	008_00
Produkt-Typ:	Management
Version:	0.2
Klassifizierung:	frei zugänglich
Verteiler:	n/a
Autor(en):	Nicolas Lanquetin, HdM
Verantwortliche(r):	Nicolas Lanquetin, HdM
Kunde:	Unipaz Europe

Issue Nr.	Autor	Verantwortlicher	Erfasst am	Letzte Änderung	Beschreibung	Status
E1	Nicolas Lanquetin	Nicolas Lanquetin	2/12/2005	2/1/2006	Externer Mitarbeiter kann über die Weihnachtszeit nicht am Projekt mitarbeiten.	geschlossen
T1	Nicolas Lanquetin	Nicolas Lanquetin	22/11/2005	26/11/2005	Komplettes Refactoring der Applikation durch Änderung des Konzeption (Unbenennen der Klassen und Verbesserung des Vererbungskonzept)	geschlossen
T2	Nicolas Lanquetin	Jochen Kornau	3/12/2005	4/1/2006	Gravierende Fehler in der XHTML-Kompatibilität. Alle Templates müssen angepasst werden.	geschlossen
P1	Nicolas Lanquetin	Nicolas Lanquetin, Jochen Kornau	5/1/2006	7/1/2006	Verzögerung des Projektplans durch Prüfungsvorbereitungen.	geschlossen

Abbildung 1.9.: Issue-Log

Abbildungsverzeichnis

8.1. Produktstrukturplan	38
8.2. Produktflussdiagramm	41
1.1. Database ERM	49
1.2. PSF UML	52
1.3. Gantt-Chart - Legende	56
1.4. Gantt-Chart - Seite 1	57
1.5. Gantt-Chart - Seite 2	58
1.6. Gantt-Chart - Seite 3	59
1.7. Gantt-Chart - Seite 4	60
1.8. Risiko-Log	63
1.9. Issue-Log	67

Tabellenverzeichnis

3.1. Auflistung der wichtigsten Aspekte für die Projektansatz Entscheidung . .	10
4.1. PS Business Case - Nicht-finanzieller Nutzen	12
3.1. Auflistung der wichtigsten Aspekte für die Projektansatz Entscheidung . .	22
4.1. Kontaktinformationen der Beteiligten	25
5.1. PS Business Case - Nicht-finanzieller Nutzen	26
6.1. Projekt Resultate / Produkte	28
6.2. Produktbeschreibung - PS Framework 1.0	28
6.3. Produktbeschreibung - Schulung und Einarbeitung	29
6.4. Produktbeschreibung - Software	29
6.5. Produktbeschreibung - Dokumentation	30
6.6. Produktbeschreibung - Umgebung	30
7.1. Kommunikationsplan	34
1.1. Notfallplan	64

Quellcodeverzeichnis

1.1. ini Konfigurationsdatei	44
1.2. ini-Datei: Konfiguration für Logger	46
1.3. ini-Datei: Konfiguration für DBUtils	47

Abkürzungsverzeichnis

CMS	C ontent M anagement S ystem
CSS	C ascading S tyle S heet
ERM	E ntity R elationship M odell
HTML	H yper T ext M arkup L anguage
LAMP	L inux A pache M ySQL P HP
LDAP	L ightweight D irectory A ccess P rotocol
PEAR	P HP E xtension and A pplication R epository
PFD	P roduct F luss D iagramm
PHP	P HP H ypertext P reprocessor
PID	P roject I nitiation D ocument
PRA	P roposal R isk A nalysis
PSF	P S Framework
PSP	P rodukt S truktur P lan
sog.	s ogenannte
URL	U niform R esource L ocator
usw.	u nd s o w eiter
WYSIWYG	W hat Y ou S ee I s W hat Y ou G et
XHTML	E Xtensible H yper T ext M arkup L anguage
z.B.	z um B eispiel