

Projekthandbuch

Jeremias Eppler, Jochen Morent, Georgi Georgiev

7. Januar 2014



Inhaltsverzeichnis

1	Projektstart	3
1.1	Projekt-Kontext-Beziehungen	3
1.1.1	Zeitlicher Kontext	3
1.1.2	Sachlicher Kontext	4
1.1.3	Sozialer Kontext	5
1.2	Ziele und Objekte	6
1.3	Projektorganisation	7
1.3.1	Projektorganigramm	7
1.4	Projektleistungsplan	8
1.4.1	Projektstrukturplan	8
1.4.2	Arbeitspakete Spezifikation	9
1.5	Terminplan	16
1.5.1	Projekt-Ressourcen-Zeitplan	16
1.5.2	Projektmeilensteinplan	17
1.5.3	Projektbalkenplan	18
2	Projektcontrolling	19
2.1	Projektfortschrittsbericht	19
2.2	Projekt-Score-Card	21
3	Projektabschluss	22
3.1	Projektabschlussbericht	22

1 Projektstart

1.1 Projekt-Kontext-Beziehungen

1.1.1 Zeitlicher Kontext

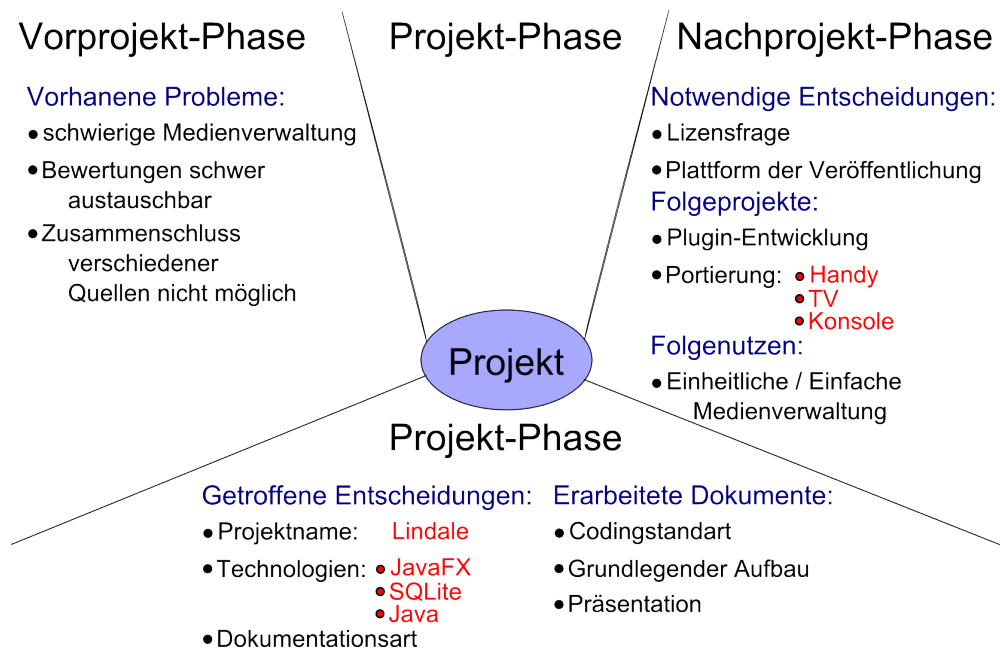


Abbildung 1.1: Zeitlicher Kontext

1.1.2 Sachlicher Kontext

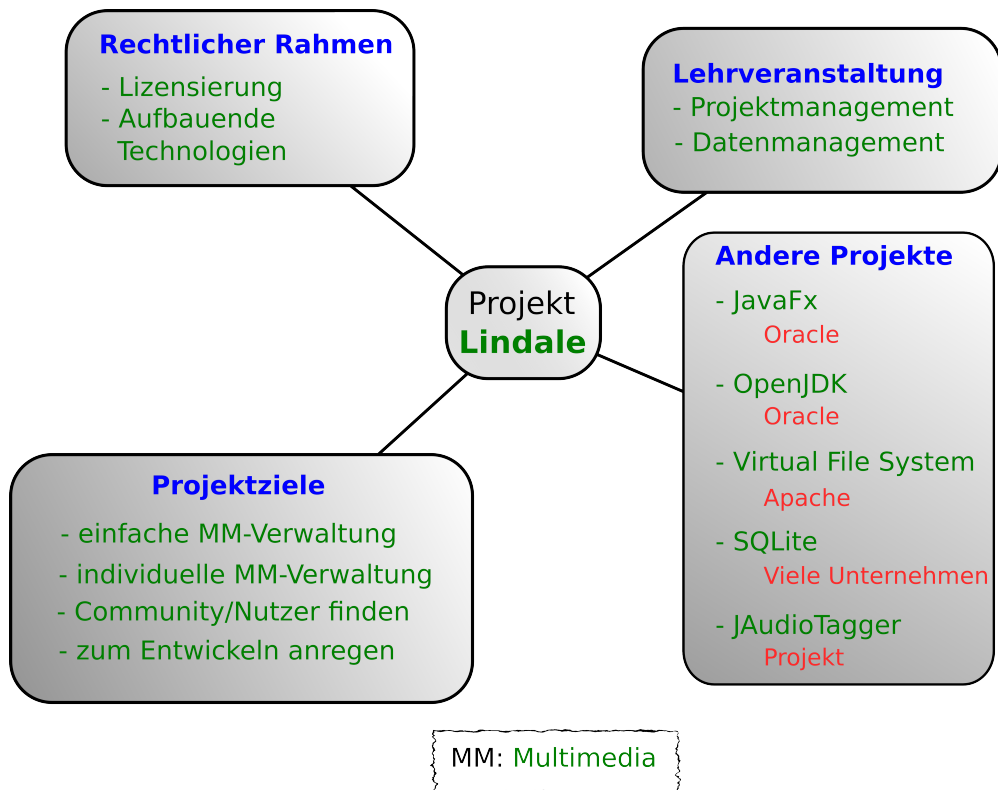


Abbildung 1.2: Sachlicher Kontext

1.1.3 Sozialer Kontext

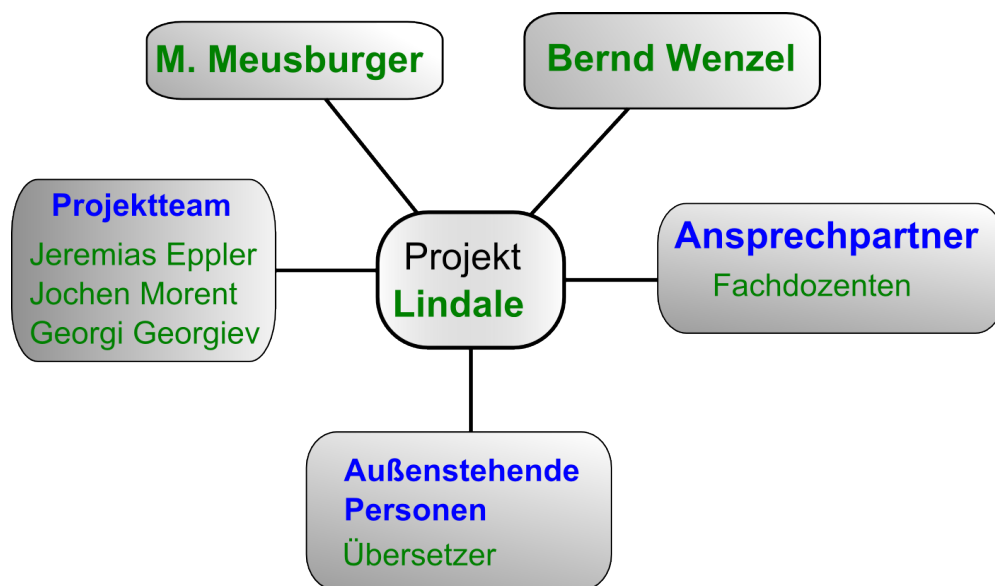


Abbildung 1.3: Sozialer Kontext

1.2 Ziele und Objekte

Hauptziele

- Verbessertes Medien-Verwaltungs-Tool
- Vereinheitlichte Medienverwaltung

Nebenziele

- Bewertungssystem
- Arbeiten mit Datenbanken
- Einarbeiten in JavaFX
- Arbeiten mit APIs

Nichtziele

- Keine Backup-Lösung
- Weder logische noch physische Speicherung wird verwaltet

Objekte

- Projektdokumentation
- Datenbankentwurf (Datenbank/Modelierung)
- UML-Diagramme (Klasse/Sequenz)
- Plugin-Manager (Manager/PluginAPI)
- Plugins
- APIs
- Ressourcen-Sammlung
- GUI-Entwurf/-Modell

1.3 Projektorganisation

1.3.1 Projektorganigramm

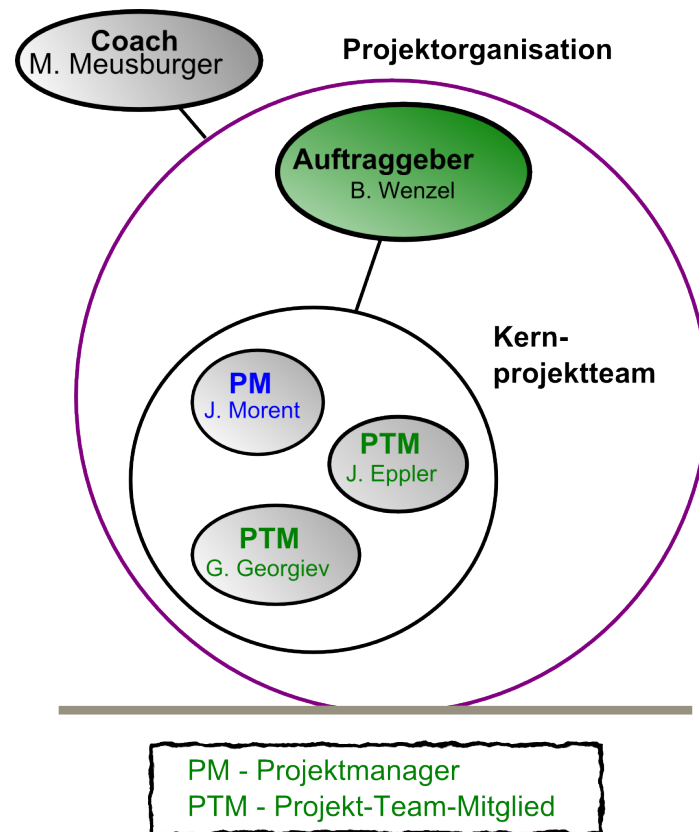


Abbildung 1.4: Projektorganisation

1.4 Projektleistungsplan

1.4.1 Projektstrukturplan

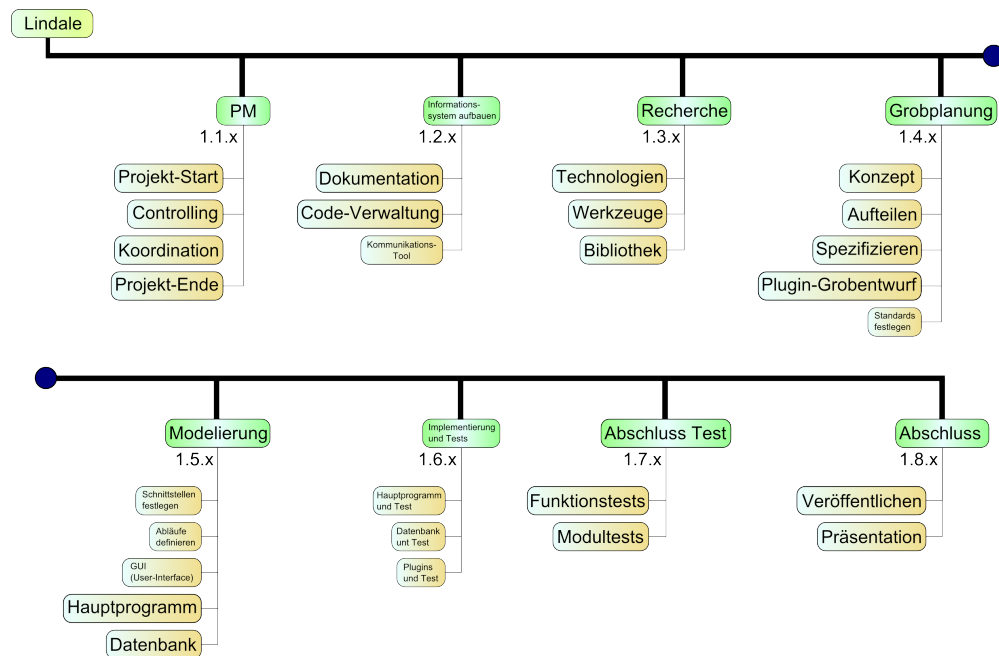


Abbildung 1.5: Projektstrukturplan

1.4.2 Arbeitspakete Spezifikation

Informationssystem aufbauen

Dokumentation (1.2.1)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none">• Linksammlung• Projektdokumentation• Protokolle• Dateien, die während des Projekts erstellt werden, sammeln• Recherche Ergebnisse
Ergebnisse	<ul style="list-style-type: none">• Durchgehende Dokumentation• Einheitlicher Informationsstand
Code-Verwaltung (1.2.2)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none">• Code Versionsverwaltung (mit GIT)• Commiten / Teamwork sammeln• Recherche Ergebnisse
Ergebnisse	<ul style="list-style-type: none">• Separat an verschiedenen oder gleichen Codestücken arbeiten• Nicht beeinflussen durch separate Entwicklungsschritte
Kommunikations-Tool (1.2.3)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none">• Facebook-Gruppe• Facebook-Gruppen-Chat
Ergebnisse	<ul style="list-style-type: none">• Absprachen, Treffen, Neuigkeiten• Schneller Informationsaustausch

Recherche

Technologien (1.3.1)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • Abschätzen des Implementierungsaufwand • Moderner Technologien Einsatz (FXML, JavaFX)
Ergebnisse	<ul style="list-style-type: none"> • Effiziente Implementierung • Richtiger Einsatz der Technologie
Werkzeuge (1.3.2)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • Ermitteln von nützlichen Tools • Ausprobieren der Werkzeuge • Dokumentieren • Auswahltreffen
Ergebnisse	<ul style="list-style-type: none"> • Erfahrungen sammeln • Geeignete Tools finden
Bibliothek (1.3.3)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • Suchen • Experimentieren • Auswahltreffen
Ergebnisse	<ul style="list-style-type: none"> • Erleichterte Programmierung • Richtiger Einsatz der Bibliotheken

Grobplanung

Aufteilung (1.4.1)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • Aufsplitten der Module <ul style="list-style-type: none"> – Startprogramm – GUI – Plugin-Manager – Datenbank – Plugins
Ergebnisse	<ul style="list-style-type: none"> • Leichtere Verteilung der Aufgaben
Spezifizieren (1.4.2)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • Tests definieren • API / Plugin-API • SQL-Wrapper
Ergebnisse	<ul style="list-style-type: none"> • Einheitliche Schnittstellen
Plugin-Grobentwurf (1.4.3)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • Arten definieren • Start-/Stop-/Instalations-/Deinstalations- Verhalten definieren • Modi (Automatisch/Manuel) • Zugriffe / Zugriffsarten • Bereitstellung
Ergebnisse	<ul style="list-style-type: none"> • Modulare Architektur
Standards festlegen (1.4.4)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • Coding-Standards • Sprach-Standards • Formatierungs-Standards
Ergebnisse	<ul style="list-style-type: none"> • Vereinheitlichung der Entwicklung

Modellierung

Schnittstellen festlegen (1.5.1)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • UML (Klassen-Diagramm) Modellierung • APIs Implementierungs nah definieren <ul style="list-style-type: none"> – Plugin-API – Interne API <ul style="list-style-type: none"> * SQL-Execution * GUI • Testfälle definieren
Ergebnisse	<ul style="list-style-type: none"> • Implementierungsnahes Model • Testdefinitionen
Abläufe definieren (1.5.2)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • UML (Sequenz-Diagramm) • Flussdiagramme • Programmstart • Plugin: Start-/Stop-/Instalations-/Deinstalations-Verhalten • Quellen Hinzufügen • Datenbankzugriffe (Wrapper ↔ SQLite) • Medien laden / abspielen • Laden von Plugins in die GUI • Fehlermeldungen
Ergebnisse	<ul style="list-style-type: none"> • Definierter Programmablauf

GUI (1.5.3)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • Grafische Modellierung • Dialoge Modellieren / Festlegen • Aussehen (Farbe, Steuerelemente, Schnittstellen zum User)
Ergebnisse	<ul style="list-style-type: none"> • Strukturierte und moderne Oberfläche
Hauptprogramm (1.5.4)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • UML (Klassen-Diagramme) • Schnittstellen zu Bibliotheken • Schnittstellen zur Datenbank • Statement-Exekution-Parser
Ergebnisse	<ul style="list-style-type: none"> • Ein Model des Hauptprogrammes
Datenbank (1.5.5)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • E3R-Methode • 5. Normalform • Seminar Wenzel • Test-Transaktionen definieren
Ergebnisse	<ul style="list-style-type: none"> • Datenbankmodel nahe Implementierung • Testdefinitionen

Implementierung und Tests

Hauptprogramm und Test (1.6.1)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • Umsetzen der Klassen-Diagramm (Dummys) • Umsetzung definierter Test (JUnit) • Implementierung der Funktionalität • GUI-Test
Ergebnisse	<ul style="list-style-type: none"> • Lauffähiges Hauptprogramm (nicht mehr grabbeln)
Datenbank und Test (1.6.2)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • Implementieren • Abfragen Tests / Transaktions Tests
Ergebnisse	<ul style="list-style-type: none"> • Super Datenbank
Plugins und Test (1.6.3)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • Implementieren • Testen • Schnittstellen Tests
Ergebnisse	<ul style="list-style-type: none"> • Funktionierende Plugins

Abschluss Test

Funktionstests (1.7.1)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • Alle Funktionalitäten überprüfen • Fehler provozieren
Ergebnisse	<ul style="list-style-type: none"> • Programm kann veröffentlicht werden
Modultests (1.7.2))	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • Jedes Modul separatem Test unterzogen • Modulgruppen testen • Teil-Programm-Tests z.B.: Hauptprogramm
Ergebnisse	<ul style="list-style-type: none"> • Funktionalität der Programmteile gewährleistet

Abschluss

Veröffentlichen (1.8.1)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • Lizenzfrage klären • Webpräsenz • Webhoster finden
Ergebnisse	<ul style="list-style-type: none"> • Veröffentlichen
Präsentieren (1.8.2)	
Vorgehensweise / Inhalte	<ul style="list-style-type: none"> • Projekthandbuch • Präsentation vorbereiten / üben / halten
Ergebnisse	<ul style="list-style-type: none"> • Präsentations Note

1.5 Terminplan

1.5.1 Projekt-Ressourcen-Zeitplan

Tabelle 1.1: Projekt-Ressourcen-Zeitplan

Projekt-Ressourcen-Zeitplan				
PSP-Code	Arbeitspaket	Jochen Morent	Jeremias Eppler	Georgi Georgiev
1.2	Informationssystem aufbauen			
1.2.1	Dokumentation	5	5	5
1.2.2	Code Verwaltung	3		
1.2.3	Kommunikationstool		3	
1.3	Recherche			
1.3.1	Technologien	5	5	5
1.3.2	Werkzeuge	5	5	5
1.3.3	Bibliothek	5	5	5
1.4	Grobplanung			
1.4.1	Konzept	2	2	2
1.4.2	Aufteilen	1	1	1
1.4.3	Spezifizieren	5	5	5
1.4.4	Plugin-Grobentwurf	5	5	5
1.4.5	Standards festlegen	1		
1.5	Modelierung			
1.5.1	Schnittstellen festlegen	3	2	2
1.5.2	Abläufe definieren	2	2	2
1.5.3	GUI	5	5	6
1.5.4	Hauptprogramm	4	3	3
1.5.5	Datenbank	6	6	6
1.6	Implementierung und Tests			
1.6.1	Hauptprogramm und Test	10	10	10
1.6.2	Datenbank und Test	3	3	
1.6.3	Plugins und Test	12	12	12
1.7	Abschlusstest			
1.7.1	Funktionstest	6	6	6
1.7.2	Modultest	8	8	8
1.8	Abschluss			
1.8.1	Veröffentlichen	10	10	10
1.8.1	Präsentation	2	2	2

1.5.2 Projektmeilensteinplan

Tabelle 1.2: Projektmeilensteinplan

Projektmeilensteinplan			
PSP-Code	Meilenstein	Plantermin	Isttermin
	Projektstart	05.10.2013	05.10.2013
1.3	Start Recherche	06.10.2013	06.10.2013
1.4	Grobplanung	05.10.2013	05.10.2013
1.4.1	Konzeptions beginn	21.11.2013	21.11.2013
1.6	Modellierung	06.12.2013	09.12.2013
1.7.1	Start Hauptprogramm Entwicklung	07.12.2013	15.12.2013
1.7.2	Beginn Plugin-Entwicklung	21.12.2013	
1.8	Start Testen	06.12.2013	
1.9	Projektabschluss	10.01.2013	

1.5.3 Projektbalkenplan

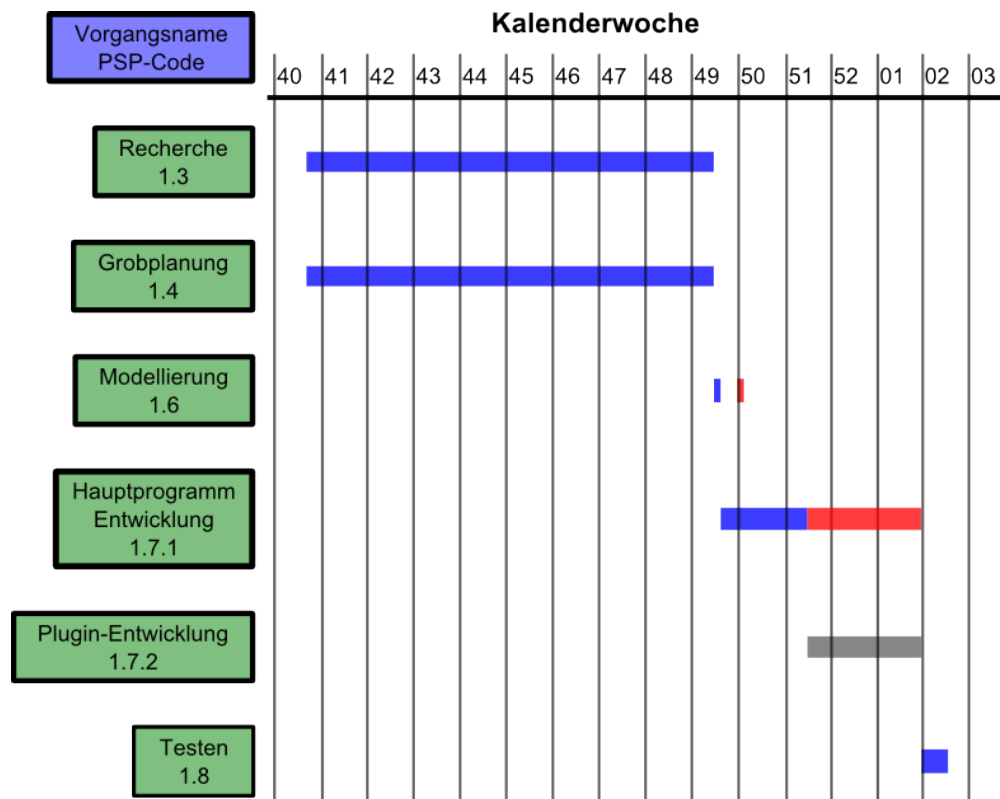


Abbildung 1.6: Projektbalkenplan

2 Projektcontrolling

2.1 Projektfortschrittsbericht

Lindale Projektfortschrittsbericht				
sehr gut	gut ✓	okay	schlecht	sehr schlecht
Gestamtstatus des Projektes				
<ul style="list-style-type: none"> • Projekt liegt im Zeitrahmen • Hinzugekommenes Know-how durch Datenbankprogrammierungs Lehrveranstaltung und Seminar • Angespannte Arbeitssituation aller Teammitglieder • Verteilung von Inhalten des Projekthandbuches → Informations und Datenmisstand innerhalb des Projektes 				
Status: Projektziele und Betrachtungsobjekte				
<ul style="list-style-type: none"> • Zieladaptierung: weglassen von Features wie: <ul style="list-style-type: none"> – Logische und Physikalische Collection – Bewertungssystem • Adaptierung der Nebenziele war notwendig dennoch → Status: Gut 				
Status: Projektleistungsfortschritt				

<ul style="list-style-type: none"> Arbeitspakete fortschritt: <ul style="list-style-type: none"> 1.2 Erfolgreich umgesetzt 1.3 Recherche für die Hauptprogramm Entwicklung abgeschlossen 1.4.3 Spezifizieren für die Entwicklung des Hauptprogramms abgeschlossen 1.4.5 Standards soweit festgelegt, das entwickelt werden kann 1.5.5 Datenbank Modellierung abgeschlossen Verzug mit dem modellieren der GUI des Hauptprogrammes → Status: Okay
Status: Projekttermine
<ul style="list-style-type: none"> Verzögerung bei der Entwicklung und Modellierung des Hauptprogramms durch neu hinzuge- wonnenes Wissen Wegen Verzögerung beim modellieren → Status: Okay
Status: Projektumwelt, Beziehungen zu anderen Projekten
<ul style="list-style-type: none"> Hinzugekommener Dozent als Experte zwischen Datenbank und Anwendungsprogramm Starke Wissenserweiterung → Status: Sehr Gut
Maßnahmenkatalog
<ul style="list-style-type: none"> Adaptieren der Nebenziele Neu ausrichtung des Projektes mithilfe des gewonnenen Informationen Zusammenführen der Projekthandbuch Informationen Einpfelegen von Informationen in das Informationssystem Information aller Teammitglieder und Umsetzung läuft → Status: Sehr Gut
Status: Anhang
<ul style="list-style-type: none"> Projekt Score Card
Version 1.1 Datum 6 Dez, 2013 Ersteller: Lindale-Team

2.2 Projekt-Score-Card



Abbildung 2.1: Projekt-Score-Card

3 Projektabschluss

3.1 Projektabschlussbericht

Projektabschlussbericht				
sehr gut	gut ✓	okay	schlecht	sehr schlecht
Gesamteindruck				
<ul style="list-style-type: none">• Projektziele nicht alle erreicht<ul style="list-style-type: none">– Grobplanung und Modellierung abgeschlossen– Technologien ausgewählt– Entwicklung des Hauptprogramms erreicht• Noch nicht erreichte Ziele, wegen Verschiebung einer Klausur<ul style="list-style-type: none">– Entwicklung von Plugins– Testen der gesamten Applikation				
Reflexion: Erfüllung der geplanten Leistungen, Einhaltung der geplanten Termine				
<ul style="list-style-type: none">• Erreichte Meilensteine<ul style="list-style-type: none">– Grobplanung und Modellierung abgeschlossen– Technologien ausgewählt– Entwicklung des Hauptprogramms mit Verspätung erreicht• Erreichte Leistungen<ul style="list-style-type: none">– viel Wissen durch ausgiebige Recherche erarbeitet– Gute Planung und modellierung der Applikation– Teilweise Implementierung des Projektes				
Reflexion: Projektumweltbeziehungen, Beziehungen zu anderen Projekten				

<ul style="list-style-type: none"> • Projektumweltbeziehung <ul style="list-style-type: none"> – Die Anzahl der Dozenten hat sich im Verlauf des Projektes von zwei Ansprechpartner auf vier Ansprechpartner erhöht • Beziehung zu anderen Projekten <ul style="list-style-type: none"> – Die Entwicklung von Java 8 durch das OpenJDK Projekt verläuft Planmäßig
Reflexion: Teamarbeit, Einsatz von Projektmanagement
<ul style="list-style-type: none"> • hoher Einsatz von Projektmanagement Methoden durch Lehrveranstaltung zum Projektmanagement • Projekt unter permanenter Kontrolle aller Teammitglieder • Gute Teamarbeit und Koordinierung durch <ul style="list-style-type: none"> – Projektteamleiter – Einsatz von Kommunikationstechnologie – klare Terminregelungen • hoher Wissensaustausch über eigenes Wiki • ausgeglichene Aufgabenverteilung auf alle Mitglieder • insgesamt sehr gute Teamarbeit
Zusammenfassende Erfahrungen für andere Projekte
<ul style="list-style-type: none"> • lernen von Projektmanagement Methoden anhand des Projektes kann in Zukunft genutzt werden • Erfahrung mit sehr neuen, teilweise noch im Beta-Stadium befindlicher, Technologien • Datenbank modellierung und programmierung wurde durch LV's erarbeitet • kennen lernen von Git als Code-Versionierungs-Verwaltung für Teams • erste Erfahrung mit Software Engeeniering von größeren Anwendungen
Version 1.0 Datum 06 Jan, 2014 Ersteller: Lindale-Team