Semesterarbeit

Projektverwaltung

B-TIN-22-T-a

Standort: Bern

Eingereicht von

Kedros Matthias

Bernstrasse 2, 3175 Flamatt

Matthias.kedros@edu.teko.ch

077 471 80 52

Eingereicht am 17.03.2025

Versionskontrolle

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Datum | Autor/in | Änderungen/Kommentare |
|  |  |  |  |
|  |  |  |  |

Management Summary

<Kurze Zusammenfassung des Inhalts des Dokuments, max. 1 Seite>

Inhaltsverzeichnis

[1. Abkürzungsverzeichnis 5](#_Toc192085843)

[2. Abbildungsverzeichnis 5](#_Toc192085844)

[3. Tabellenverzeichnis 5](#_Toc192085845)

[4. Projektinitialisierung 5](#_Toc192085846)

[Ausgangslage 5](#_Toc192085847)

[Rahmenbedingungen 6](#_Toc192085848)

[Prozessbezogene Rahmenbedingungen 6](#_Toc192085849)

[Produktbezogene Rahmenbedingungen 7](#_Toc192085850)

[Abgrenzung 7](#_Toc192085851)

[Stakeholder-Analyse 8](#_Toc192085852)

[Ziele 9](#_Toc192085853)

[Lösungskonzept mit Varianten und Beurteilung 10](#_Toc192085854)

[Varianten 10](#_Toc192085855)

[Machbarbeitsbeurteilung 11](#_Toc192085856)

[Variantenentscheid 11](#_Toc192085857)

[Projektmanagement 12](#_Toc192085858)

[Projektplanung 12](#_Toc192085859)

[Detailliert mit Diagramm 12](#_Toc192085860)

[Risikoanalyse 13](#_Toc192085861)

[Qualitätsmanagement 14](#_Toc192085862)

[Testkonzept 14](#_Toc192085863)

[Konfigurationsmanagement 15](#_Toc192085864)

[Konzept 16](#_Toc192085865)

[Kontextdiagramm 16](#_Toc192085866)

[Geschäftsprozessanalyse 17](#_Toc192085867)

[Geschäftsprozesse 17](#_Toc192085868)

[Aktivitätsdiagramme 18](#_Toc192085869)

[Detailanforderungen 19](#_Toc192085870)

[Funktionale Anforderungen 19](#_Toc192085871)

[Nicht-funktionale Anforderungen 20](#_Toc192085872)

[Use-Case-Beschreibungen 21](#_Toc192085873)

[Sequenzdiagramme 22](#_Toc192085874)

[Projekt Anlegen 22](#_Toc192085875)

[Fortschritt erfassen 22](#_Toc192085876)

[Modellierung der Klassen 23](#_Toc192085877)

[Klassendiagramm 23](#_Toc192085878)

[Beschreibung der Fachklassen 24](#_Toc192085879)

[Zustandsdiagramme 24](#_Toc192085880)

[Systemarchitektur 24](#_Toc192085881)

[Testkonzept 24](#_Toc192085882)

[GUI-Design 24](#_Toc192085883)

[5. Realisierung 24](#_Toc192085884)

[Programmierumgebung / Programmierrichtlinien 24](#_Toc192085885)

[Softwareaufbau 24](#_Toc192085886)

[GUI-Implementierung 24](#_Toc192085887)

[Datenbankimplementierung und -anbindung 24](#_Toc192085888)

[Testprotokoll 24](#_Toc192085889)

[6. Fazit / Lessons learned 24](#_Toc192085890)

[7. Eigenständigkeitserklärung 25](#_Toc192085891)

[8. Glossar 25](#_Toc192085892)

[9. Anhang 25](#_Toc192085893)

# Abkürzungsverzeichnis

|  |  |
| --- | --- |
| Abkürzung | Bedeutung |
|  |  |
|  |  |
|  |  |

# Abbildungsverzeichnis

<...>

# Tabellenverzeichnis

<...>

# Projektinitialisierung

## Ausgangslage

Die Firma benötigt eine Applikation, mit der sich ihre Projekte auf einfache Weise verwalten lassen. Dabei sollen sowohl die Planungsdaten (z. B. geplante Start- und Endtermine, Budget) als auch die effektiv erfassten Werte (tatsächlich benötigte Zeit, entstandene Kosten) in einem zentralen System hinterlegt werden können. Auf diese Weise erhalten Projektleiter\*innen einen umfassenden Überblick über ihre laufenden und abgeschlossenen Projekte und verfügen zugleich über eine solide Basis für ein effizientes Projektreporting. Darüber hinaus müssen Projektmitarbeitende die Möglichkeit haben, Dokumente direkt in der Applikation zu referenzieren, um die gemeinsame Ablage von projektbezogenen Informationen zu vereinfachen.

Zunächst wird auf Grundlage der vorliegenden Informationen ein Lastenheft erstellt. Dazu sind alle vorhandenen Anforderungen zu prüfen und eventuell bestehende Lücken zu schliessen. Anschliessend dient das fertige Lastenheft als Grundlage für die Entwicklung eines Prototyps des geplanten Informatiksystems. Dabei soll das System sowohl die Projektplanung als auch die Dokumentation im laufenden Betrieb abbilden können, um den gesamten Projektlebenszyklus übersichtlich und effizient zu unterstütze

## Rahmenbedingungen

### Prozessbezogene Rahmenbedingungen

Das Projekt wird als Informatikprojekt im Rahmen einer Semesterarbeit durchgeführt und folgt dem Wasserfallmodell. Dabei ist von Beginn an sicherzustellen, dass sämtliche im Studium erlernten Methoden und Instrumente des Projektmanagements sowie des System Engineerings konsequent in die Praxis umgesetzt werden. Dies umfasst eine fundierte und detaillierte Planung aller Aktivitäten, ein kontinuierliches Projektcontrolling, eine sorgfältige Risikoanalyse mit konkreten Gegenmassnahmen, ein angemessenes Qualitätsmanagement sowie ein klar strukturiertes Konfigurationsmanagement. Analyse und Design werden mittels UML-Diagramme dokumentiert, um eine eindeutige und nachvollziehbare Kommunikationsbasis zu schaffen. Sämtliche Schritte im Projekt müssen dabei so gestaltet werden, dass sie den Anforderungen an eine vollständige und transparente Projektdokumentation genügen.

### Produktbezogene Rahmenbedingungen

Die zu entwickelnde Applikation muss unterschiedliche sequenzielle Vorgehensmodelle unterstützen und es ermöglichen, Projekte eindeutig in Phasen und Aktivitäten zu gliedern. Für jede Phase und Aktivität sollen sowohl geplante als auch effektive Werte (z. B. Termine, Kosten, Ressourcen) verwaltet werden können. Zudem ist sicherzustellen, dass eine komfortable Verwaltung und Verknüpfung von Dokumenten – sowohl innerhalb der Applikation als auch mit externen Systemen oder Ablagen – möglich sind. Die Struktur des Systems muss darüber hinaus eine flexible Erweiterung für zukünftige Anforderungen gewährleisten und dabei den Datenschutz- und Sicherheitsaspekten in allen Bereichen gerecht werden.

## Abgrenzung

Ziel des Projekts ist die Entwicklung einer Applikation, mit der sich Projekte effizient planen und verwalten lassen. Dies umfasst insbesondere die Erfassung und Auswertung von Plan- und Ist-Daten (z. B. Termine, Ressourcen, externe Kosten) sowie das Referenzieren zugehöriger Dokumente. Dabei stehen folgende Aspekte im Vordergrund:

1. **Planungs- und Projektmanagement-Funktionen**
   * Anlage, Bearbeitung und Überwachung von Projekten, Projektphasen und Aktivitäten gemäss einem in der Applikation hinterlegten Vorgehensmodell.
   * Verwaltung von Terminen (Start- und Enddaten, Meilensteine) und Fortschrittsangaben (Phasenfortschritt, Aktivitätsfortschritt).
   * Hinterlegen und Abrufen von relevanten Dokumenten auf Projekt-, Phasen- oder Aktivitätsebene.
2. **Projektphase vs. Produktivsystem**
   * Der Schwerpunkt liegt auf der Erstellung eines **Prototyps**. Dieser dient dazu, die grundlegenden Funktionen und Abläufe aufzuzeigen und zu testen.
   * Der Produktiveinsatz in einer grossflächigen, unternehmensweiten Umgebung und die Einbindung in ein umfassendes Unternehmensumfeld (z. B. verteilte Systeme, Single Sign-on oder automatisierte Schnittstellen) wird **nicht** im Rahmen dieser Semesterarbeit realisiert.
3. **Technische Abgrenzung**
   * Die Implementierung erfolgt in einer überschaubaren technischen Umgebung (z.B. lokale oder Testdatenbank, einfaches GUI).
   * Hochverfügbarkeits- und Skalierungskonzepte, Automatisierung von Deployment-Prozessen sowie weiterführende Sicherheitsmassnahmen (wie ein eigenes Rollen- und Rechtemanagementsystem) werden **nicht** im Detail ausgearbeitet.

## Ein Bild, das Text, Screenshot, Schrift, Rechteck enthält. KI-generierte Inhalte können fehlerhaft sein.Stakeholder-Analyse

Bei der Bewertung der Stakeholder wird unterschieden, wie gross ihr Einfluss (verticale Achse) auf das Projekt ist und wie stark ihr Interesse (horizontale Achse) an den Projektergebnissen ausgeprägt ist. Daraus ergeben sich vier Quadranten, in denen jede Gruppe jeweils eine eigene Managementstrategie erfordert:

**Manage closely**

Im ersten Quadranten „Manage closely“ ist der Projektleiter angesiedelt. Aufgrund seiner Entscheidungskompetenzen und seines fortwährenden Engagements für die Zielerreichung nimmt er eine Schlüsselstellung ein. Da er gleichermassen einen hohen Einfluss wie ein grosses Interesse am Projekterfolg hat, muss ein enger Austausch sichergestellt werden. Das bedeutet regelmässige Abstimmungen, schnelle Freigaben und eine zeitnahe Kommunikation zu jeder relevanten Änderung oder Risikosituation.

**Keep satisfied**

Im zweiten Quadranten „Keep satisfied“ finden sich die Geschäftsleitung und die Finanzabteilung. Beide haben grosse Auswirkungen auf Budget, strategische Ausrichtung und Wirtschaftlichkeit des Vorhabens, legen jedoch den Schwerpunkt primär auf Kennzahlen, Statusberichte und übergreifende Projektergebnisse. Aufgrund des eher geringeren Interesses an operativen Details ist es sinnvoll, sie vor allem mit kompakten und aufschlussreichen Zusammenfassungen zu versorgen. So erhalten sie genau die Informationen, die sie für ihre Entscheidungen benötigen, ohne vom Tagesgeschäft überlastet zu werden.

**Keep informed**

Im dritten Quadranten „Keep informed“ sind die Projektmitarbeitenden und die Endnutzer einzuordnen. Sie besitzen ein starkes Interesse an der praktischen Umsetzung und am Resultat, können jedoch nur bedingt formale Entscheidungen treffen oder den Projektverlauf massgeblich beeinflussen. Es ist daher wichtig, ihnen regelmässig Einblicke in den aktuellen Stand, die anstehenden Aufgaben und mögliche Veränderungen zu geben, um ihre Motivation hochzuhalten und ihre fachliche Expertise gezielt einzubringen. Auf diesem Wege entsteht eine offene Kommunikation, die Fehlentwicklungen vorbeugt und gleichzeitig zur Akzeptanz des Endprodukts beiträgt.

**Monitor**

Im vierten Quadranten „Monitor“ befindet sich die IT-Abteilung, die zwar die nötige Infrastruktur bereitstellt, insgesamt aber weder einen unmittelbaren Einfluss auf grundlegende Projektentscheidungen ausübt noch ein gesteigertes Interesse an fortlaufenden Details hat. Sie sollte immer dann einbezogen werden, wenn konkrete technische Anforderungen, Wartungsfenster oder Installationen anstehen. Ansonsten reicht es aus, sie bei relevanten Meilensteinen kurz zu informieren, sodass im Falle einer Anpassung schnell reagiert werden kann.

## Ziele

Ziel dieser Projektarbeit ist es, eine schlanke und dennoch funktionale Applikation zur Projektverwaltung zu entwickeln, die sämtliche Phasen und Aktivitäten eines Projekts von der Planung bis zum Abschluss übersichtlich abbildet. Dabei sollen sowohl geplante als auch tatsächlich erfasste Daten (z. B. Termine, Kosten, benötigte personelle Ressourcen) erfasst und miteinander verglichen werden können. Ein zentrales Anliegen ist es, Projektleitenden einen schnellen und klaren Überblick über ihre laufenden und abgeschlossenen Projekte zu verschaffen und so ein effizientes Projektreporting zu ermöglichen. Gleichzeitig sollen Mitarbeitende in der Lage sein, Dokumente direkt im System zu referenzieren, um Doppelspurigkeiten zu vermeiden und eine konsistente Ablage sicherzustellen. Da es sich um eine Semesterarbeit mit begrenztem Zeitrahmen handelt, wird das System bewusst als Prototyp mit minimalem Funktionsumfang umgesetzt. Dieses Vorgehen stellt sicher, dass alle wesentlichen Anforderungen erfüllt werden, ohne den Fokus auf eine saubere, leicht nachvollziehbare und dokumentierte Lösung zu verlieren.

## Lösungskonzept mit Varianten und Beurteilung

### Varianten

**Variante 1: Website mit Datenbank**   
Die Applikation wird als eigenständige Webanwendung mit einer Datenbank entwickelt. Das Backend regelt die Geschäftslogik, das Frontend stellt die Benutzeroberfläche bereit. **Vorteile**  
Hohe Flexibilität, gute Erweiterbarkeit und eine leistungsfähige Verwaltung von Daten. **Nachteile**  
Erfordert mehr Fachwissen, höheren Installationsaufwand und eine durchdachte Serverkonfiguration. **Einsatzszenario**  
Geeignet für Lösungen mit komplexen Anforderungen und parallelem Zugriff vieler Nutzer.

**Variante 2: Microsoft Forms mit Datenbank**   
Formulare werden mit Microsoft Forms erstellt, die eingegebenen Daten laufen in eine Datenbank. Ein automatisierter Ablauf übermittelt die Informationen kontinuierlich.  
**Vorteile**  
Schnelle Formulardefinition, unkomplizierte Datenerfassung und nahtlose Integration in bestehende Umgebungen.  
**Nachteile**  
Geringe Flexibilität im Design und Mehraufwand bei der Konfiguration der Datenbankanbindung.  
**Einsatzszenario**  
Ideal für schnelle Prototypen, bei denen ein Teil der Daten strukturiert in einer zentralen Datenbank vorliegen soll.

**Variante 3: Microsoft Forms mit Serialisierung**  
Daten werden in Microsoft Forms erfasst und anschliessend in Dateien gespeichert. Ein automatisierter Export erzeugt Einträge im gewünschten Format.  
**Vorteile**  
Sehr einfach einzurichten, kein Datenbankserver und minimaler Konfigurationsaufwand.  
**Nachteile**  
Kein direkter Mehrbenutzerbetrieb und eingeschränkte Möglichkeit für Abfragen und Berichte.  
**Einsatzszenario**  
Sinnvoll für kleine, zeitlich begrenzte Prototypen oder einfache Datensammlungen ohne komplexe Auswertungen.

### Machbarbeitsbeurteilung

**Variante 1: Website mit Datenbank**

Technische Machbarkeit: Gut umsetzbar bei vorhandener Webserver- und Datenbankinfrastruktur und entsprechenden Kenntnissen.

Organisatorische Machbarkeit: Höherer Koordinationsaufwand für Serverbetrieb und Wartung.

Aufwand: Programmierung, Serverkonfiguration und Datenbankdesign sind zeit- und ressourcenintensiver als bei den anderen Varianten.

**Variante 2: Microsoft Forms mit Datenbank**

Technische Machbarkeit: Schnell umsetzbar, wenn Microsoft 365 verfügbar ist und die Datenbank unkompliziert angebunden werden kann.

Organisatorische Machbarkeit: Wenig Koordinationsaufwand, da Microsoft Forms einfach zu handhaben ist.

Aufwand: Formulare anlegen und Datenbankanbindung konfigurieren sind überschaubar, jedoch weniger flexibel.

**Variante 3: Microsoft Forms mit Serialisierung**

Technische Machbarkeit: Sehr leicht realisierbar, da keine Datenbankinstallation erforderlich ist.

Organisatorische Machbarkeit: Minimaler Aufwand, solange die Datenmengen klein bleiben und keine komplexen Auswertungen nötig sind.

Aufwand: Wenig Implementierungsarbeit, aber manuelle oder halbautomatische Verarbeitung der Daten (Exportdateien) kann bei grösserem Umfang aufwändig werden.

### Variantenentscheid

Da dies ein Schulprojekt ist und mir die Erfahrung im eigenständigen Erstellen einer Website fehlt, entscheide ich mich für **Variante 2**. Diese Lösung erfordert nur wenig technisches Know-how, ist mit minimalem Aufwand umsetzbar und eignet sich gut, um rasch einen funktionsfähigen Prototyp zu erstellen. Wäre mehr Zeit und technisches Fachwissen vorhanden, würde die Website mit Datenbank Variante vermutlich die langfristig bessere Option darstellen.

## Projektmanagement

### Projektplanung

Das Projekt folgt dem Wasserfallmodell, wobei die Phasen nacheinander durchlaufen und jeweils aufeinander aufbauen.

Von KW 4 bis KW 5 werden im Rahmen der Projektinitialisierung zuerst Ausgangslage und Rahmenbedingungen geklärt sowie Stakeholder- und Anforderungsanalysen durchgeführt (teilweise bis KW 6).

Darauf folgt in KW 7 und KW 8 die Konzeptphase, in der UML-Diagramme erstellt, Prozesse analysiert, Detailanforderungen beschrieben und das Datenbankdesign erarbeitet werden.

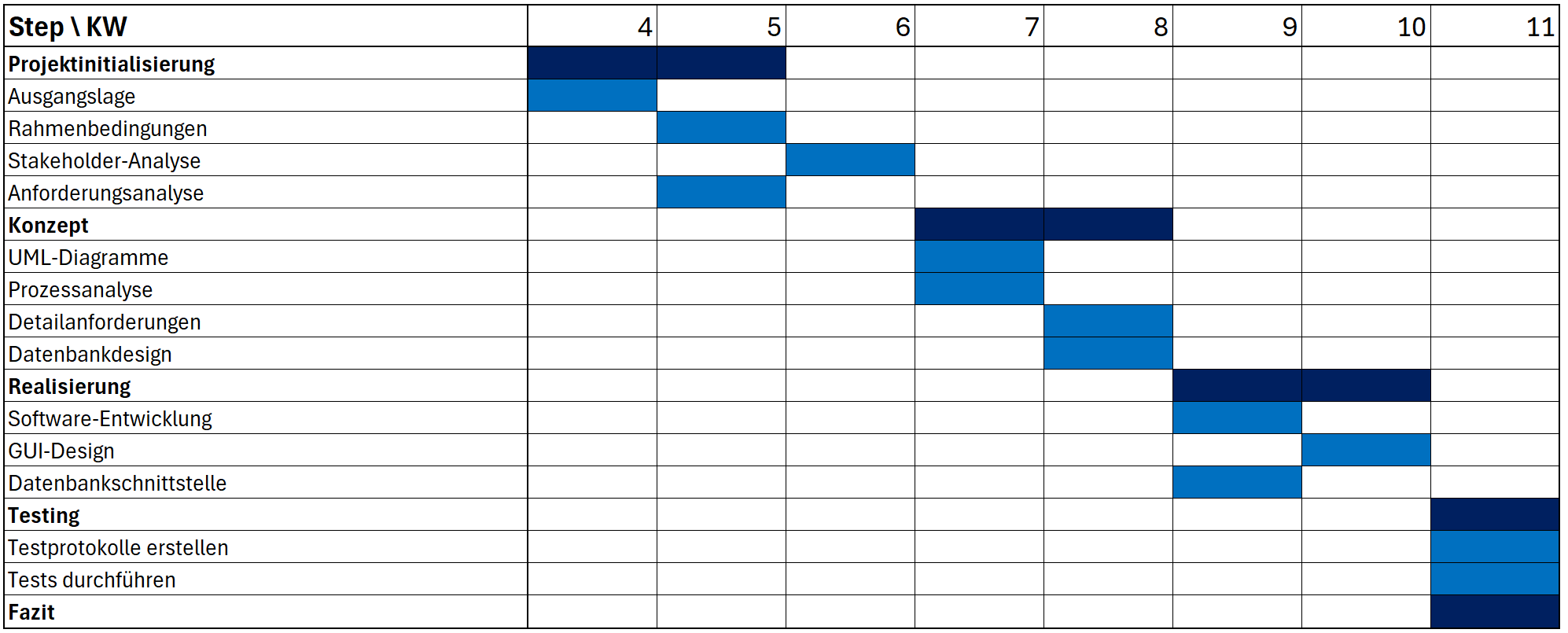
In der Realisierungsphase (KW 8–KW 9) entstehen die Software-Kernfunktionen, das GUI wird gestaltet und die Datenbankschnittstelle integriert.

Anschliessend beinhaltet die Testing-Phase (KW 10) das Erstellen von Testprotokollen und die Durchführung der System- und Abnahmetests.

Das Fazit wird schliesslich in KW 11 gezogen, um das Projekt abzuschliessen und die gewonnenen Erkenntnisse zu dokumentieren.

### Detailliert mit Diagramm

Soll:



Ist:

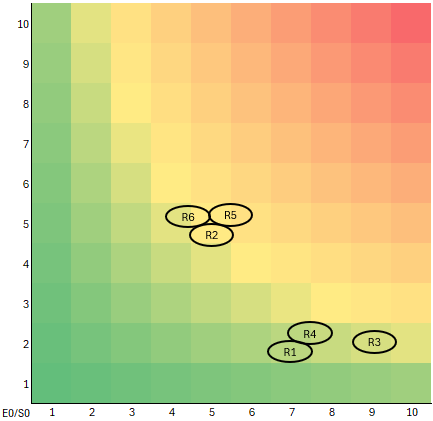
<Wird am Ende eingefügt>

### Risikoanalyse

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Im Rahmen des Projekts wurde ein Risikokatalog erstellt, um potenzielle Gefahren frühzeitig zu erkennen und geeignete Massnahmen festzulegen. Die Tabelle zeigt das jeweilige Risiko (inkl. Einschätzung zu Eintrittswahrscheinlichkeit und Schadensausmass), die daraus resultierenden Risikowerte sowie mögliche Gegenmassnahmen und einen Umsetzungsentscheid. Grundsätzlich sollten in einem realen Projekt alle aufgeführten Massnahmen konsequent angewendet werden, um das Risiko auf ein Minimum zu reduzieren. Da es sich hier jedoch um ein Schulprojekt mit begrenztem Umfang handelt, wurden nur ausgewählte Massnahmen umgesetzt oder weiterverfolgt. Diese Vorgehensweise ermöglicht einerseits, die wichtigsten Risiken zu adressieren, und schont andererseits die verfügbaren Ressourcen.



### Qualitätsmanagement

Da es sich um ein schulisches Projekt mit Prototypcharakter handelt, beschränkt sich das Qualitätsmanagement auf ein Minimum. Ziel ist, grobe Fehler möglichst frühzeitig zu vermeiden, ohne dabei umfangreiche Ressourcen in formale Qualitätskontrollen zu investieren.

Grundprinzipien

* Einfachheit: Schlanke Prozesse und möglichst wenige formale Kontrollen.
* Frühzeitige Entdeckung von Fehlern: Kurze, informelle Reviews in frühen Phasen, um gröbste Mängel rasch zu erkennen.
* Pragmatischer Test: Kurzer Funktions- und Abnahmetest, um die wichtigsten Anwendungsfälle zu überprüfen.

### Testkonzept

Das Testkonzept soll sicherstellen, dass der entwickelte Prototyp die wichtigsten Anforderungen erfüllt und gravierende Fehler möglichst frühzeitig entdeckt und behoben werden. In einem echten Projekt würde dieses Konzept umfassender ausfallen und sich stärker an den gängigen Vorgehensmodellen orientieren. Dabei werden üblicherweise zunächst die Anforderungen analysiert und daraus die wichtigsten Testfälle abgeleitet, bevor eine detaillierte Planung der Testdurchführung erfolgt. Im nächsten Schritt werden die Testumgebung und Testdaten vorbereitet, um anschließend gezielte Unit-Tests und Integrations-Tests durchzuführen. Danach folgt der Systemtest, der das gesamte Zusammenspiel der Komponenten prüft und klärt, ob sämtliche Anforderungen wie erwartet umgesetzt sind. Die Testergebnisse werden dokumentiert, gefundene Fehler protokolliert und priorisiert, um sie möglichst schnell zu beheben. Zum Abschluss führen die Stakeholder oder der Auftraggeber den Abnahmetest durch, bei dem sie das System aus Anwendersicht bewerten und über die Freigabe entscheiden. In einem professionellen Umfeld würde man zusätzlich automatisierte Tests und spezielle Werkzeuge einbeziehen sowie eine kontinuierliche Integration mit automatisierten Builds und Testläufen etablieren. Auf diese Weise ließe sich die Qualität und Stabilität des Produkts noch stärker absichern. In diesem Projekt mit Prototypcharakter genügt dagegen ein schlanker Ansatz: Die Testaktivitäten werden weitgehend manuell durchgeführt, und das Hauptaugenmerk liegt auf dem zügigen Auffinden kritischer Probleme sowie dem Nachweis, dass die Kernfunktionen lauffähig sind. So entsteht ein ausgewogenes Verhältnis zwischen Aufwand und Nutzen, das den Rahmenbedingungen eines schulischen Projekts gerecht wird.

### Konfigurationsmanagement

Auch das Konfigurationsmanagement wird bei diesem schulischen Prototyp-Projekt bewusst minimal gehalten. Ziel ist es, den Überblick über den Entwicklungsstand zu behalten und Anpassungen nachvollziehen zu können, ohne umfangreiche Verfahren oder Werkzeuge einzusetzen.

**Ziele**

* **Versionierung**: Sicherstellen, dass alle Projektartefakte (Quellcode, Dokumentation) in einer nachvollziehbaren Versionsverwaltung liegen.
* **Nachvollziehbarkeit**: Änderungen an zentralen Dokumenten und Quellcode sollen jederzeit rückverfolgbar sein
* **Minimale Verwaltung**: Möglichst geringer Aufwand durch einfache Tools und Prozesse.

**Konfigurationsobjekte**

* **Quellcode**: Die im Projekt entstehenden Klassen, Skripte und Konfigurationsdateien.
* **Dokumentation**: Projektbericht, UML-Diagramme, Testprotokolle etc.

In diesem Prototyp-Projekt genügt ein Git-Repository als zentrales Tool, um Versionsstände zu speichern und Änderungen transparent zu machen.

#### Wie ist git konfiguriert

Um das Konfigurationsmanagement in diesem schulischen Prototyp-Projekt möglichst schlank zu halten, wird Git als zentrales Versionierungswerkzeug eingesetzt. Nach der Installation von Git auf dem jeweiligen Entwicklungsrechner sollten zunächst die globalen Benutzerdaten eingetragen werden, damit alle Commits klar zugeordnet werden können. Dies erfolgt über die Befehle

git config --global user.name "<Benutzername>"  
git config --global user.email "<E-Mail-Adresse>"

Anschließend lässt sich im Projektordner mit

git init

ein lokales Git-Repository anlegen mit

git clone <Repository-URL>

klonen. Um ein eigenes Remote-Repository als „Source of Truth“ hinzuzufügen, wird folgende Anweisung verwendet:

git remote add origin <Repository-URL>

Zusätzlich wird eine Datei namens „.gitignore“ erstellet, um temporäre, generierte oder vertrauliche Dateien vom Versionsmanagement auszuschließen. Bei jedem Commit sollte eine kurze, aussagekräftige Commit-Nachricht erfasst werden, damit Änderungen später leicht nachverfolgbar sind. Für ein schulisches Projekt mit Prototypcharakter genügt üblicherweise eine einfache Branching-Strategie, regelmäßige Commits und eine klare Struktur bei der Ablage der Dokumente und des Quellcodes. Auf diese Weise ist sichergestellt, dass alle Projektmitglieder stets den aktuellen Stand einsehen können, wichtige Meilensteine reproduzierbar bleiben und Änderungen transparent dokumentiert sind.

### Konzept

## Kontextdiagramm

Ein Bild, das Text, Diagramm, Entwurf, Zeichnung enthält.

KI-generierte Inhalte können fehlerhaft sein.

Das Kontextdiagramm verdeutlicht, wie die Projektverwaltungs-Applikation mit den beteiligten Akteuren und Systemen interagiert. Der Projektleiter legt neue Projekte an, plant Phasen und Aktivitäten und überwacht deren Budgets und Fortschritte. Projektmitarbeitende erfassen Arbeitszeiten, Dokumente und Kosten, während die Geschäftsleitung Projektstatus und Berichte abruft. Das Personalsystem stellt dafür aktuelle Daten zu Mitarbeitenden bereit, und das Finanzsystem ermöglicht den Abgleich zwischen geplantem und effektivem Budget.

## Geschäftsprozessanalyse

Im Rahmen dieser Dokumentation wurden einige relevante Geschäftsprozesse in Tabellenform sowie durch Aktivitätsdiagramme dargestellt und beschrieben. Da der Projektumfang begrenzt ist, wurde auf eine vollständige Modellierung sämtlicher Prozesse verzichtet. Stattdessen liegt der Fokus auf den wichtigsten und aussagekräftigsten Prozessen, um eine Übersicht der Geschäftsprozesse zu erhalten.

### Geschäftsprozesse

Projektverwaltung



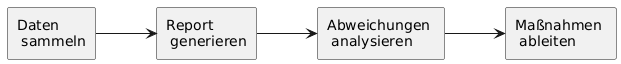
Hierbei geht es um das Anlegen, Bearbeiten und Abschliessen von Projekten. Zu Beginn werden die grundlegenden Projektdaten hinterlegt. Anschliessend können Status und weitere Informationen wie Beteiligte, Priorität oder Dokumente aktualisiert werden. Ist das Projektziel erreicht, wird das Projekt offiziell abgeschlossen und archiviert.

Ressourcen- und Kostenmanagement:



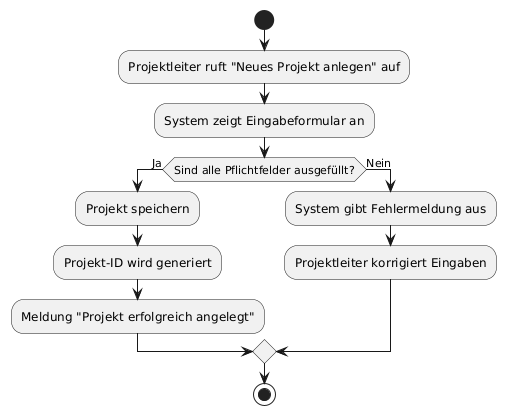
In diesem Prozess werden personelle und finanzielle Ressourcen geplant und überwacht. Zunächst werden Mitarbeitende und ihre Funktionen im System erfasst. Anschliessend erfolgt die Zuweisung von Ressourcen zu einzelnen Phasen oder Aktivitäten. Ausserdem werden externe Kosten dokumentiert, budgetiert und bei Bedarf Anpassungen vorgenommen, um Abweichungen frühzeitig zu erkennen.

Reporting und Controlling

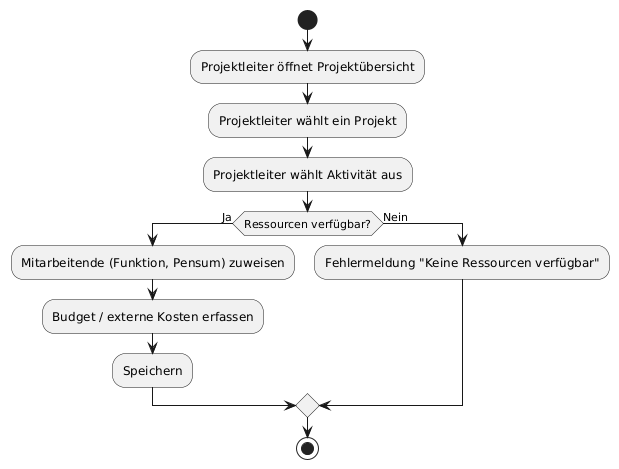


Dieser Prozess dient dazu, den Fortschritt und die Wirtschaftlichkeit der Projekte laufend zu überwachen. Dazu sammelt das System relevante Daten und bereitet sie in Berichten auf. Nach der Analyse von Abweichungen in Budget oder Zeitplan werden notwendige Massnahmen abgeleitet und umgesetzt, um eine effektive Steuerung und ein rechtzeitiges Eingreifen zu gewährleisten.

### Aktivitätsdiagramme

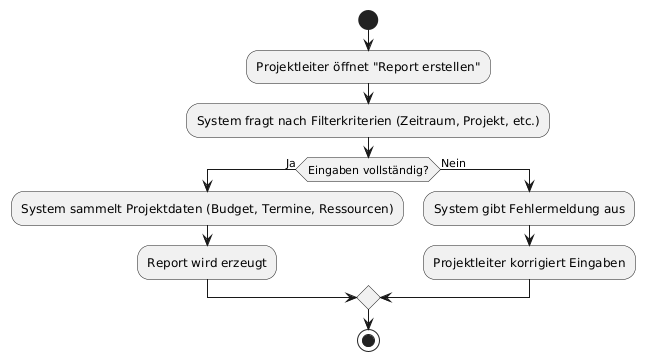
**Neues Projekt wird angelegt**

Dieser Ablauf beschreibt das Anlegen eines neuen Projekts in der Applikation. Der Projektleiter öffnet zunächst das Formular „Neues Projekt anlegen“ und trägt alle erforderlichen Projektdaten ein. Das System validiert die Eingaben und weist gegebenenfalls auf fehlende Pflichtfelder hin. Sind alle Angaben korrekt, wird das Projekt mit einer eindeutigen ID im System gespeichert, und es erfolgt eine Bestätigung, dass die Anlage erfolgreich war.

**Ressourcen zuweisen**

Hier wird dargestellt, wie personelle Ressourcen einem Projekt oder einer Aktivität zugeordnet werden. Zuerst wählt der Projektleiter ein vorhandenes Projekt und die zu bearbeitende Aktivität aus. Anschliessend werden verfügbare Mitarbeiter oder Funktionen mit dem entsprechenden Pensum erfasst. Wenn alle Daten vollständig sind und noch Kapazitäten vorhanden sind, wird die Zuweisung bestätigt und im System gespeichert.

**Fortschrittsreport generieren**

Dieser Ablauf zeigt, wie das System einen Report über den Projektfortschritt bereitstellt. Nach der Eingabe von Filterkriterien sammelt die Anwendung alle relevanten Informationen wie Budget, Termine und Ressourcen. Anschliessend erstellt sie automatisch einen Report. Bei unvollständigen Eingaben oder fehlenden Daten gibt das System eine Fehlermeldung aus, sodass die Eingaben zunächst korrigiert werden müssen.

## Detailanforderungen

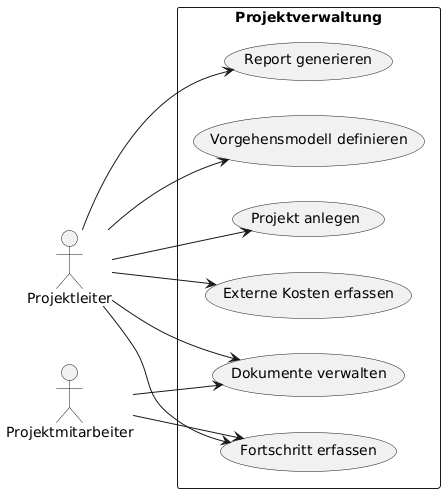
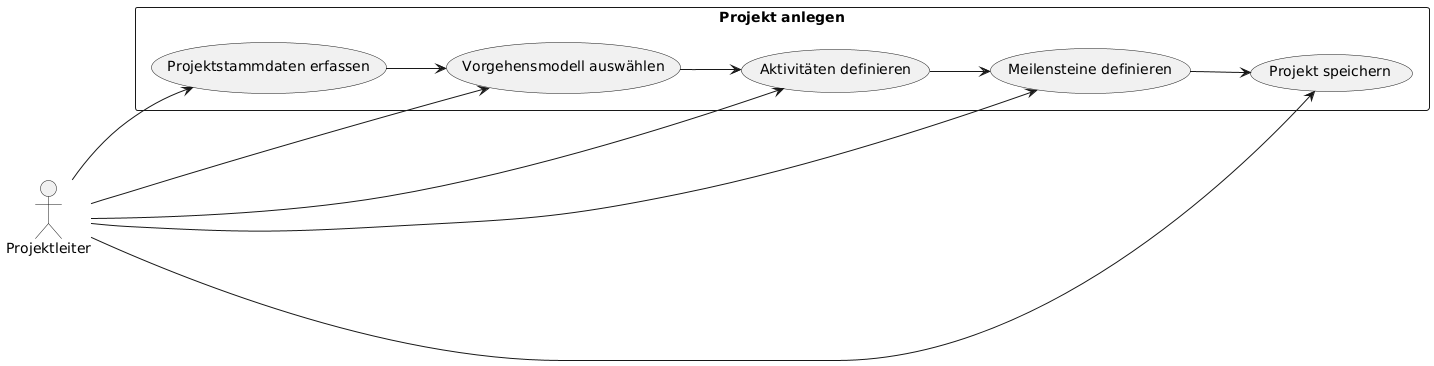
### Funktionale Anforderungen

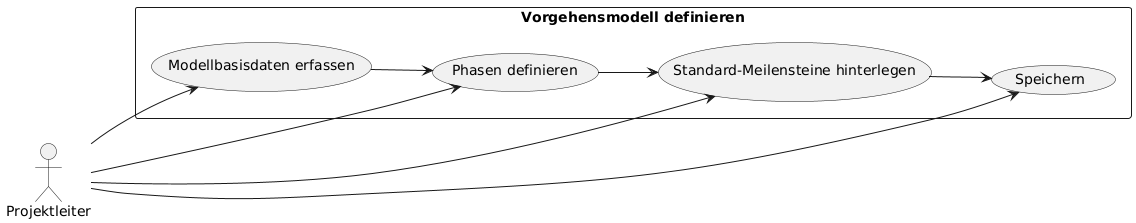
1. **Projektverwaltung**
   * Das System muss das Anlegen und Bearbeiten von Projekten ermöglichen.
   * Für jedes Projekt sollen zentrale Informationen wie Projektreferenz, Projekttitel, Projektbeschreibung sowie geplante und effektive Start- und Enddaten verwaltet werden können.
   * Der aktuelle Status eines Projekts, wie z. B. „in Planung“, „laufend“, „abgeschlossen“, muss jederzeit nachvollziehbar sein.
2. **Vorgehensmodell hinterlegen**
   * Das System muss unterschiedliche sequenzielle Vorgehensmodelle aufnehmen können.
   * Jedes Vorgehensmodell ist in Phasen und Aktivitäten unterteilt, damit Projekte nach diesem Modell strukturiert werden können.
3. **Phasen- und Aktivitätenmanagement**
   * Zu jeder Projektphase müssen geplante und effektive Zeiträume, ein Status und ein Phasenfortschritt erfasst werden.
   * Zu jeder Aktivität müssen geplante und effektive Aufwände sowie Verantwortlichkeiten hinterlegt werden.
   * Für jede Aktivität können externe Kosten in geplanten und effektiven Werten erfasst werden.
4. **Ressourcen- und Mitarbeiterverwaltung**
   * Das System muss Stammdaten zu Mitarbeiter führen können.
   * Bei der Zuteilung einer Person zu einer Aktivität müssen Rolle/Funktion, geplante Zeit und effektiv verbrauchte Zeit erfasst werden.
5. **Meilensteine**
   * Zu jeder Projektphase müssen mindestens ein Abschlussmeilenstein sowie optional zusätzliche Meilensteine definiert werden können.
   * Jeder Meilenstein ist mit einem Status (offen, erreicht, verschoben) und einem geplanten sowie effektiven Erreichungsdatum zu hinterlegen.
6. **Dokumentenmanagement**
   * Dokumente können entweder dem Projekt insgesamt, einer einzelnen Phase oder einer Aktivität zugeordnet werden.
   * Das System soll den Speicherort oder einen Referenzlink zum Dokument ablegen und diese Verknüpfung sichtbar machen.

### Nicht-funktionale Anforderungen

1. **Benutzerfreundlichkeit**
   * Die Benutzeroberfläche muss intuitiv bedienbar sein. Die wichtigsten Funktionen sind klar strukturiert und mit aussagekräftigen Bezeichnungen versehen.
2. **Wartbarkeit**
   * Die Anwendung soll so implementiert sein, dass Erweiterungen ohne grossen Aufwand vorgenommen werden können.
   * Quellcode und Dokumentation müssen klar strukturiert sein und den im Projekt definierten Konventionen entsprechen.
3. **Performanz**
   * Eingaben und Aktualisierungen sollen ohne merkliche Verzögerung verarbeitet werden.

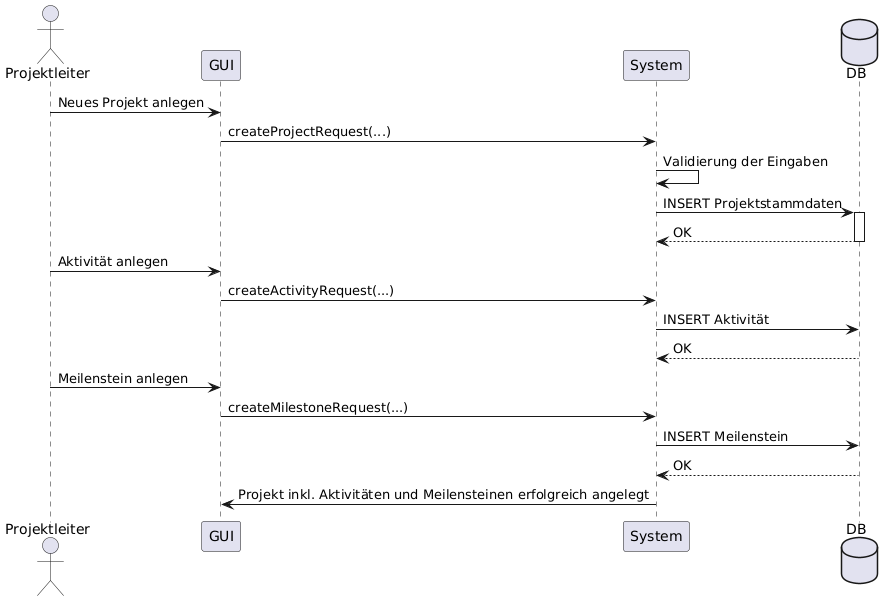
## Use-Case-Beschreibungen



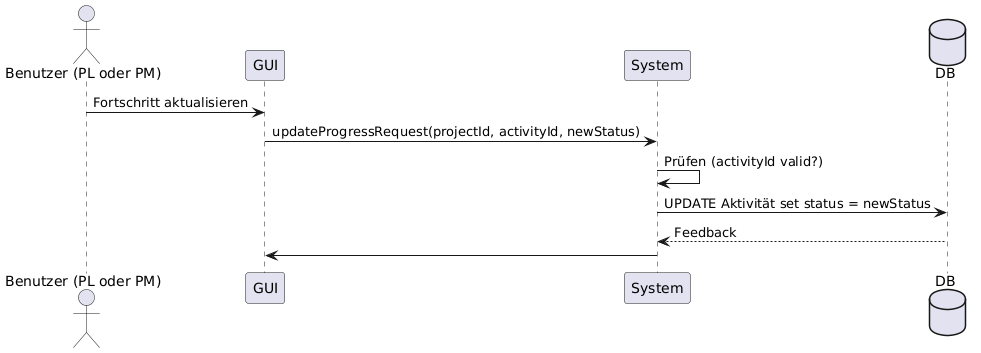


## Sequenzdiagramme

### Projekt Anlegen



### Fortschritt erfassen



## Modellierung der Klassen

### Klassendiagramm

Ein Bild, das Text, Diagramm, Plan, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.

### Beschreibung der Fachklassen

**Projekt**

* **erstelleProjekt(String titel, String beschreibung, Date bewilligungsdatum, String prioritaet)**  
  Diese Methode erstellt ein neues Projekt mit den grundlegenden Metadaten wie Titel, Beschreibung, Bewilligungsdatum und Priorität. Sie initialisiert außerdem die zugehörigen Projektphasen basierend auf dem gewählten Vorgehensmodell.
* **aktualisiereProjektstatus(String status)**  
  Diese Methode aktualisiert den aktuellen Status des Projekts, z. B. "In Planung", "In Bearbeitung" oder "Abgeschlossen".
* **berechneProjektfortschritt()**  
  Der Fortschritt des Projekts wird auf Basis des Fortschritts aller zugehörigen Phasen und Aktivitäten berechnet. Dabei wird der Durchschnitt der abgeschlossenen Phasen und Aktivitäten als Gesamtfortschritt ermittelt.
* **fuegeDokumentHinzu(Dokument dokument)**  
  Diese Methode fügt ein neues Dokument zum Projekt hinzu, sodass es zentral gespeichert und referenziert werden kann.
* **setzeVorgehensmodell(Vorgehensmodell modell)**  
  Legt das Vorgehensmodell für das Projekt fest und generiert automatisch die notwendigen Phasen entsprechend dem Modell.

**Projektphase**

* **erstellePhase(Date startdatum, Date enddatum)**  
  Erstellt eine neue Projektphase mit den entsprechenden Start- und Enddaten. Diese Phase wird dem zugehörigen Projekt zugewiesen.
* **aktualisierePhasenstatus(String status)**  
  Aktualisiert den Status der Phase, beispielsweise "Geplant", "In Bearbeitung" oder "Abgeschlossen".
* **berechnePhasenfortschritt()**  
  Der Fortschritt der Phase wird durch den Durchschnitt der bereits abgeschlossenen Aktivitäten innerhalb dieser Phase berechnet.
* **fuegeMeilensteinHinzu(Meilenstein meilenstein)**  
  Fügt einen neuen Meilenstein zur Phase hinzu, um wichtige Ereignisse innerhalb des Projektverlaufs zu markieren.
* **fuegeDokumentHinzu(Dokument dokument)**  
  Ermöglicht das Hinzufügen eines Dokuments, das spezifisch zur Projektphase gehört.

**Aktivitaet**

* **erstelleAktivitaet(Date startdatum, Date enddatum, Mitarbeiter verantwortlicher)**  
  Erstellt eine neue Aktivität mit geplanten Start- und Enddaten sowie einer verantwortlichen Person, die für die Durchführung dieser Aktivität zuständig ist.
* **aktualisiereAktivitaetsstatus(String status)**  
  Ändert den aktuellen Status der Aktivität, z. B. "Nicht gestartet", "In Bearbeitung" oder "Abgeschlossen".
* **berechneAktivitaetsfortschritt()**  
  Berechnet den Fortschritt der Aktivität basierend auf abgeschlossenen Aufgaben oder Meilensteinen.
* **weiseRessourceZu(PersonelleRessource ressource)**  
  Weist eine oder mehrere personelle Ressourcen einer Aktivität zu, um festzulegen, wer an der Aktivität beteiligt ist und mit welchem Zeitbudget.
* **erfasseExterneKosten(ExterneKosten kosten)**  
  Erfasst externe Kosten für eine Aktivität, beispielsweise für Software-Lizenzen oder Beratungsdienstleistungen.
* **fuegeDokumentHinzu(Dokument dokument)**  
  Ermöglicht das Hochladen und Verknüpfen von Dokumenten mit der Aktivität, z. B. Arbeitsanweisungen oder Ergebnisse.

**Mitarbeiter**

* **aktualisierePensum(float neuesPensum)**  
  Ändert das Arbeitspensum eines Mitarbeiters, falls sich seine Verfügbarkeit für das Projekt ändert.
* **setzeRollen(List<String> rollen)**  
  Aktualisiert die Rollen des Mitarbeiters innerhalb des Projekts, z. B. "Projektleiter", "Entwickler" oder "Tester".

**ExterneKosten**

* **setzeKosten(float budget, float kostenEffektiv)**  
  Setzt das geplante Budget für externe Kosten und erfasst die tatsächlichen Kosten, die für externe Dienstleistungen oder Produkte angefallen sind.
* **berechneAbweichung()**  
  Vergleicht das ursprüngliche Budget mit den tatsächlichen Kosten und gibt die Differenz zurück, um Budgetüberschreitungen zu erkennen.

**PersonelleRessource**

* **setzeZeit(float budget, float zeitEffektiv)**  
  Speichert die geplante und tatsächliche Arbeitszeit für eine Ressource, um Abweichungen analysieren zu können.
* **berechneAbweichung()**  
  Berechnet die Abweichung zwischen den ursprünglich geplanten und tatsächlich geleisteten Arbeitsstunden.

**Meilenstein**

* **setzeDatum(Date datum)**  
  Setzt oder ändert das Datum eines Meilensteins, um den Fortschritt eines Projekts genau zu dokumentieren.

**Dokument**

* **setzePfad(String pfad)**  
  Speichert den Speicherort des Dokuments, damit es von den Teammitgliedern abgerufen werden kann.
* **setzeTyp(String typ)**  
  Definiert den Dokumenttyp (z. B. PDF, Word, Excel), um die Dateiverwaltung zu erleichtern.

## Systemarchitektur

MVC

## Testkonzept

<...>[Vorgehen, Testobjekte, Testfälle]

## GUI-Design

Bis da für hüt

# Realisierung

[projekt- und aufgabenabhängig]

## Programmierumgebung / Programmierrichtlinien

<...>

## Softwareaufbau

<...>

## GUI-Implementierung

<...>

## Datenbankimplementierung und -anbindung

[inkl. referenzielle Integritäten, allfällige Triggers, stored procedures, usw.]

<...>

## Testprotokoll

<...>

# Fazit / Lessons learned

<...>

# Eigenständigkeitserklärung

Arbeiten, die nachweisbar in vollen Umfang oder in den wesentlichen Teilen unverändert oder ohne korrekte Quellenangabe übernommen werden, gelten als vorfabriziert und werden nicht bewertet.

Ich bestätige, dass ich die vorliegende Diplomarbeit selbstständig verfasst und alle benutzten Quellen gekennzeichnet habe. Diese Arbeit wurde weder in gleicher noch in ähnlicher Form bereits einer Prüfungskommission vorgelegt.

Name / Vorname:

Kedros Matthias

Ort / Datum / Unterschrift: