WDB  
Studie

| Auftraggeber | Ninck Georg |
| --- | --- |
| Projektleiter | Häberli Joel |
| Autor | Albrecht Miro, Häberli Joel, Schor David, Keller Anuraly |
| Klassifizierung | Nicht klassifiziert, ~~Intern, Vertraulich, GEHEIM~~ |
| Status | Zur Prüfung, ~~In Arbeit, Genehmigt~~ |
|  |  |

Änderungsverzeichnis

| Datum | Version | Änderung | Autor |
| --- | --- | --- | --- |
| 29.08.17 | 0.1 | Entwurf | Schor David |
| 5.09.17 | 0.2 | Kapitel füllen | Albrecht Miro, Häberli Joel, Keller Anuraly, Schor David |
| 12.09.17 | 0.3 | Kapitel fertigstellen | Albrecht Miro, Häberli Joel, Keller Anuraly, Schor David |
| 13.09.17 | 0.4 | Kapitel zusammenfügen | Schor David |
| 13.09.17 | 1.0 | Abgabe | Schor David |

**Inhaltsverzeichnis**

1 Situationsanalyse 3

1.1 Ausgangslage 3

1.2 Stärken der IST-Situation 3

1.3 Schwächen der IST-Situation 4

2 Ziele 5

3 Nutzen 6

4 Liste der Stakeholder 6

5 Anforderungen 7

5.1 Funktionale Anforderungen 7

5.2 Nicht-Funktionale Anforderungen 8

6 Lösungsvarianten 8

6.1 Variantenübersicht 8

6.2 Beschreibung der Varianten 9

6.2.1 Standalone Desktop-Applikation 9

6.2.2 Webapplikation mit etabliertem Framework 9

6.2.3 Thin-Client und API 10

6.3 Schutzbedarfsanalyse (ISDS) 10

6.4 Mittelbedarf 11

6.5 Wirtschaftlichkeit 12

7 Bewertung der Varianten (Tabelle) 13

8 Konsequenzen 13

9 Lösungsbeschreibung und Empfehlung 14

10 Planung 15

11 Projektführung 15

11.1 Zusammenfassung 15

11.2 Projektorganisation 16

11.3 Projektberichte 16

11.4 Kommunikation / Projektmarketing 16

11.5 Qualitätssicherung 17

11.5.1 Vorgehen zur Qualitätssicherung 17

11.5.2 Qualitätsziele 18

11.5.3 Initialer Prüfplan 19

11.5.4 Prüfmethoden 20

11.5.5 Prüfspezifikationen 20

11.5.6 Checklisten für die Prüfung der Projektdokumente 20

11.5.7 Testfalltabellen 21

11.6 Konfigurationsmanagement 21

11.6.1 Konfigurationsidentifikation 21

11.6.2 Ablagestruktur 21

11.6.3 Namenskonventionen 22

12 Projektantrag 22

# Situationsanalyse

## Ausgangslage

Jeder Informatiklernende muss eine Individuelle Praxisarbeit (IPA) absolvieren, um seine Lehre erfolgreich abzuschliessen. Um uns bereits jetzt auf diesen anspruchsvollen Abschluss vorzubereiten, möchten wir Gelerntes und Erfahrungen zentral dokumentieren. An der IPA sollten möglichst viele Informationen schnell verfügbar sein.

Unser Problem ist jetzt, in welcher Form wir diese Dokumentation umsetzen wollen. Im Moment dokumentieren wir auf Papier, in Word-Dokumenten oder auch gar nicht. Alle drei Varianten eignen sich in kleinem Rahmen - mit zunehmender Informationsmenge werden jedoch alle unbrauchbar.

Jetzt wollen wir eine Wissensdatenbank realisieren, die wir in der Zeit bis zur IPA mit Informationen füllen können. Es muss möglich sein, Informationen dem System einfach hinzuzufügen und ebenso schnell wieder auszulesen.

## Stärken der IST-Situation

Eine wichtige Stärke der bestehenden Situation ist das schnelle und einfache Dokumentieren der Arbeit in einer Text-Datei. Ebenso ist es ein grosser Vorteil, dass noch keine solche Software existiert, was bedeutet, dass es nichts zu verbessern gibt. Somit sind wir sehr flexibel und müssen keine Rücksicht auf Bestehendes nehmen.

## Schwächen der IST-Situation

Folgend werden entscheidende Schwächen der IST-Situation aufgelistet:

|  |  |  |
| --- | --- | --- |
| Nr. | Schwäche | Verbesserung |
| S1 | Hier mal kurz eine Notiz schreiben, weil man kein Papier hat, oder da und noch von einem anderen Thema hier, kann beim späteren Gebrauch zu Verwirrung führen. | Durch unsere Anwendung werden Notizen schön sauber getrennt, was bei der späteren Verwendung einiges erleichtert. |
| S2 | Sollte man in einem älteren Fach Probleme mit dem Stoff gehabt haben oder möchte dies wieder auffrischen, stehen oft nur die eigenen Notizen sowie die Dokumente der Gibb zur Verfügung. | Durch die Möglichkeit, dass jeder sein Wissen und seine Notizen teilen kann, haben wir eine breitere Auswahl zu einem Thema. Dies ermöglicht uns ein einfacheres wieder und oder neu Lernen. |
| S3 | Manchmal ist es von Vorteil, wenn man zu einem Thema eine andere Ansicht betrachtet. Hierzu muss man sich mit jemanden in Verbindung setzen, fragen ob man seine Notizen bekommt und eine Austauschmöglichkeit finden. | Hier haben wir eine einheitliche Plattform, wo man die hochgeladenen Dokumente durch Stichworte finden kann, ohne dass man um 2 Uhr nachts im Stress allen schreiben muss, um an Informationen zu kommen. |
| S4 | Immer die Notizpapiere klug und einfach aufzubewahren, kann manchmal eine Herausforderung sein. | Notizen werden kategorisch und zentral abgelegt, damit es zu keinen Verwirrungen kommt. |
| S5 | In einem grossen Papierstapel ist es sehr mühsam, etwas gezielt zu finden. | Eine Suchfunktion ermöglicht, Informationen schnell zu finden. |

# Ziele

**Systemziele**

| Nr. | Kategorie | Beschreibung | Messgrösse | Priorität |
| --- | --- | --- | --- | --- |
| Z1 | Bedienung | Wir wollen Notizen schnell und einfach erfassen, indem wir die Software übersichtlich gestalten. | Dieses Ziel ist erreicht, wenn sich neue Benutzer ohne Anleitung zurecht finden. | M |
| Z2 | Nachschlagen | Wir wollen Notizen verwalten und finden, indem wir eine Suchfunktion in die Software einbauen. | Dieses Ziel ist erreicht, wenn das Suchen von Notizen möglich ist. | M |
| Z3 | Abschluss | Das Projekt ist in diesem Semester abgeschlossen. | Dieses Ziel ist erreicht, wenn das Projekt abgeschlossen wurde. | M |
| Z4 | Editor | Wir wollen, dass es einen guten Editor gibt, welcher Text formatiert und Bilder eingefügt werden können. | Dieses Ziel ist erreicht, wenn ein funktionsfähiger Editor mit den gewünschten Funktionen integriert wird. | 1 |
| Z5 | Teilen | Wir wollen, dass die Dokumentationen und Notizen einfach teilbar sind und wollen dies mit einer einfachen Teilen-Funktion und Upload-Funktion erreichen. | Dieses Ziel ist erreicht, wenn das Teilen von Notizen möglich ist. | 2 |

Legende: Priorität: M=Muss /1=hoch, 2=mittel, 3=tief

**Vorgehensziele**

| Nr. | Kategorie | Beschreibung | Messgrösse | Priorität |
| --- | --- | --- | --- | --- |
| V1 | Qualität | Qualität wird stetig sichergestellt | Vergleich: IST - SOLL | M |
| V2 | Planung | Planung wird eingehalten | Die Planung wird nach Plan eingehalten | M |
| V3 | Kommunikation | Jeden Tag ein kurzer Statusbericht | Teammeetings | M |
| V4 | Einsatzfähig | Durchgehend ein Funktions-/Einsatzfähiges Produkt | Funktionsfähig | 1 |

**Rahmenbedingungen**

Eine Rahmenbedingung für dieses Projekt ist, dass es sich um eine Neuentwicklung handelt. Da es keine ähnliche Produkte gibt, sind wir sehr flexibel, was die Funktionen angeht.  
Die Entwicklung dieses Produkts sollte in einem kurzen Zeitraum realisiert werden.

Als weitere Rahmenbedingungen muss die eingeschränkte Arbeitszeit im Modul 306 beachtet werden. Die Entwicklung dieses Produkts wird deshalb nicht konzentriert realisiert werden.

Eine weitere Rahmenbedingung ist, dass jeder Mitarbeiter einen funktionsfähigen Laptop oder Computer während der Arbeitszeit bzw. in der Gewerbeschule besitzt oder benutzen darf.

Zusätzlich muss jeder Mitarbeiter die benötigten Softwares wie Word, Compiler für die gewählte Entwicklungssprache und eine Internetverbindung zur Verfügung gestellt bekommen.

**Abgrenzung**

Dieses Projekt ist nicht als Wikipedia-Ersatz vorgesehen, denn es soll nicht nur Wissen sondern auch Erfahrungen abgespeichert werden. Dieses Projekt soll auch kein Lernmanagement-Tool sein (um Wörter zu lernen oder Hausaufgaben zu machen).

# Nutzen

Der allgemeine Nutzen ist, den Umgang mit einer grossen (hier: nicht Google- oder Wikipedia-gross, aber was man in ein paar Jahren Schule lernt) Menge Wissen beziehungsweise Gelerntem zu vereinfachen. Neues Wissen soll archiviert werden und nach Bedarf schnell wieder abgerufen werden können.

Ein spezifischer Nutzen ergibt sich an der IPA. Wenn wir vorher die Wissensdatenbank gewissenhaft füllen, können wir die benötigten Informationen schnell abrufen. Das wird nötig sein, da wir an der IPA keine Zeit haben, lange im Internet (Google, StackOverflow, etc.) nach Dingen zu suchen, die wir eigentlich bereits gelernt haben.

Zu guter Letzt sind die eigenen Erfahrungen sehr wertvoll, welche aber oft mit der Zeit vergessen gehen und schon gar nicht online verfügbar sind. Auch davon werden wir an der IPA profitieren können.

# Liste der Stakeholder

Folgend sind alle wichtige Personen für das Projektvorhaben aufgelistet:

|  |  |  |
| --- | --- | --- |
| Partner | Rolle | Person |
| Strategisch seitens Anwender und Betreiber | Auftraggeber | Ninck Georg |
| Ersteller/Anwender | Projektleiter | Häberli Joel |
| Ersteller/Anwender | Mitarbeiter/Entwickler | Albrecht Miro, Häberli Joel, Keller Anuraly, Schor David |

# Anforderungen

## Funktionale Anforderungen

Folgend sind alle funktionalen Anforderungen aufgelistet:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Ziel-Nr. | Kategorie | Anforderung | Gewicht |
| A01.a | Z1 | Funktional | Der Benutzer kann seine Notizen einfach und übersichtlich mit Titel, Inhalt, Anhang und Tags erfassen. | M |
| A01.b | Z1 | Funktional | Im Nachhinein hat der Benutzer auch die Möglichkeit, seine Notizen zu bearbeiten. | M |
| A02 | Z2 | Funktional | Der Benutzer kann seine Notizen nach Tags und Titel filtern und durchsuchen. | M |
| A03 | Z3 | Funktional | Die Wissensdatenbank ist am Ende des Semesters mit den Grundfunktionen funktionsfähig, so dass der Benutzer sie einsetzen kann. | M |
| A04 | Z4 | Funktional | Der Benutzer kann seinen Text formatieren und kann Bilder hinzufügen. | 1 |
| A05 | Z5 | Funktional | Der Benutzer kann Notizen mit anderen ohne grossen Aufwand austauschen. | 2 |

Legende: Priorität: M=Muss /1=hoch, 2=mittel, 3=tief

## Nicht-Funktionale Anforderungen

Folgend sind alle nicht funktionalen Anforderungen aufgelistet:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Ziel-Nr. | Kategorie | Anforderung | Gewicht |
| A06 | Z1 | Nicht-Funktional | **Qualität** Die Qualität der Software soll hoch sein. Diese kann mit Wartbarkeit und Code-Guidelines erhöht werden. | M |
| A07 | - | Nicht-Funktional | **Verfügbarkeit** Die Software muss immer verfügbar sein. Dies kann die Sicherheit und die Stabilität der Software deutlich erhöhen. | M |
| A08 | - | Nicht-Funktional | **Performance** Die Software soll eine hohe Performance haben. Die Wartezeiten beim Laden, Speichern oder Suchen darf nicht 2 Sekunden überschreiten. Dies kann mit Coding-Guidelines und ohne unnötig komplizierte Algorithmen verbessert werden. | M |

Legende: Priorität: M=Muss /1=hoch, 2=mittel, 3=tief

# Lösungsvarianten

## Variantenübersicht

Um die Wissensdatenbank umzusetzen, gibt es verschiedene Lösungsansätze:

* Eine Standalone Desktop-Application mit Java, die auf jedem PC lokal installiert wird
* Eine klassische Webapplikation mit etabliertem Framework. Die Webapplikation wird zentral gehostet
* Oder ein minimaler Thin-Client und eine Server-Applikation mit einer API. Die Applikation kann lokal oder extern gehostet werden

## Beschreibung der Varianten

### Standalone Desktop-Applikation

Diese Lösungsvariante besteht darin, die erforderlichen Anforderungen in einer Desktop-Applikation mit Java umzusetzen. Für die Datenspeicherung wird MySQL verwendet.

**Struktur (grobe Architektur)**

Das Backend wird durch eine MySQL-Datenbank abgebildet. Die grafische Benutzeroberfläche wird mit JavaFX realisiert. Für die Schnittstelle zwischen Datenbank und Logikschicht kann ein Framework eingesetzt werden.

**Schnittstellen**

Es sind keine Schnittstellen zu anderen Systemen erforderlich.

**Abdeckung der Anforderungen**

Die Anforderungen werden alle abgedeckt. Das Teilen von Daten kann bei dieser Variante umständlich sein.

**Realisierbarkeitsbetrachtung**

Diese Variante ist realisierbar. Technische Risiken können ausgeschlossen werden. Der Aufwand für die spätere Weiterentwicklung darf nicht vernachlässigt werden.

### Webapplikation mit etabliertem Framework

Diese Lösungsvariante besteht darin, die erforderlichen Anforderungen in einer Web-Applikation mit PHP umzusetzen. Für die Datenspeicherung wird MySQL verwendet.

**Struktur (grobe Architektur)**

Das Backend wird durch eine MySQL-Datenbank abgebildet. Die Ausgaben werden mit PHP realisiert. Dafür kann auch ein Framework eingesetzt werden. Die Webapplikation wird zentral gehostet.

**Schnittstellen**

Es sind keine Schnittstellen zu anderen Systemen erforderlich.

**Abdeckung**der Anforderungen

Die Anforderungen werden alle abgedeckt. Da alle Daten zentral gespeichert werden, ist das Teilen von Daten bei dieser Variante am einfachsten umzusetzen.

**Realisierbarkeitsbetrachtung**

Diese Variante ist realisierbar. Technische Risiken können während der Entwicklung ausgeschlossen werden. Danach muss der Betrieb entweder selbst gehostet werden oder an einen externen Dienstleister übertragen werden. Es besteht das Risiko, dass der Betrieb durch einen Zwischenfall gestört wird (*single point of failure*).

### Thin-Client und API

Diese Lösungsvariante besteht darin, die erforderlichen Anforderungen in einer Server-Applikation umzusetzen. Für die Datenspeicherung wird MySQL verwendet. Wichtige Eigenschaften dieser Variante sind, dass der Server komplett unabhängig vom (Thin)-Client ist und nur eine definierte API anbietet. Es können verschiedene Clients realisiert werden. Es wird ein webbasierter Beispiel-Client entwickelt.

**Struktur (grobe Architektur)**

Das Backend wird durch eine MySQL-Datenbank abgebildet. Um den Datenzugriff zu vereinfachen wird ein ORM-Framework eingesetzt. Die Webapplikation wird zentral oder lokal gehostet. Um die API zu realisieren wird Spring Boot verwendet.

**Schnittstellen**

Es sind keine Schnittstellen zu anderen Systemen erforderlich. Der Server bietet aber eine Schnittstelle für den Client (bzw. beliebige Clients).

**Abdeckung der Anforderungen**

Die Anforderungen werden alle abgedeckt. Daten können innerhalb einer Instanz einfach ausgetauscht werden.

**Realisierbarkeitsbetrachtung**

Diese Variante ist realisierbar. Technische Risiken können während der Entwicklung ausgeschlossen werden. Danach muss der Betrieb entweder selbst gehostet werden oder an einen externen Dienstleister übertragen werden. Es besteht das Risiko, das der Betrieb durch einen Zwischenfall gestört wird.

## Schutzbedarfsanalyse (ISDS)

Der Inhalt der Wissensdatenbank kann privat sein. Deshalb müssen die Daten sorgfältig behandelt werden. Es werden aber keine persönlichen oder sogar besonders schützenswerte Daten gespeichert.

Wichtig ist jedoch die Verfügbarkeit. Gehen die Daten (kurz vor der IPA) verloren, können betroffene Benutzer in eine sehr schwierige Situation kommen.

Bei der ersten Variante ist der Benutzer selbst für den Schutz seiner Daten zuständig, da die Daten lokal bleiben. Bei der dritten Variante kann er aktiv die Verantwortung übernehmen (selbst hosten). Trotzdem muss bei der Entwicklung der Varianten zwei und drei Wert auf Sicherheit gelegt werden. Es müssen die "Wissensdaten" und Benutzeraccounts verwaltet werden.

## Mittelbedarf

Kosten (CHF)

| Phase | Geplant |
| --- | --- |
| Initialisierung\* | 0 |
| Konzept | 0 |
| Realisierung | 0 |
| Einführung | 0 |
| Total | 0 |

\*Vorleistung (IST)

**Personalaufwand**

Es ist vorgesehen, das ganze Projekt im Modul 306 durchzuführen.

Bis zur Abgabe des Projekts stehen uns noch 9x4 Lektionen zur Verfügung bzw. 108 Personenstunden. Davon sind 18 Stunden für die Theorie. Somit stehen 90 Personenstunden für die Konzeption, Realisierung und Einführung, zur Verfügung.

| Phase | Geplant |
| --- | --- |
| Initialisierung\* | 30 |
| Konzept | 30 |
| Realisierung | 40 |
| Einführung | 20 |
| Total | 120 |

\*Vorleistung (IST)

**Sachmittel**

Der Raum wird in der GIBB von der GIBB bereitgestellt. Auch wird ein Meeting-Raum im BIT verwendet. Die IT-Infrastruktur ist vorhanden. Die Entwicklung der Software soll auch ortsunabhängig weitergeführt werden. Dazu bietet sich ein Notebook an, welches mit den benötigten Softwares konfiguriert werden muss. Wenn kein Notebook zur Verfügung steht, muss die vmWP2 für dieses Projekt konfiguriert werden.

* Windows 10
* Office
* Teamplanungssoftware
* Entwicklungsumgebung (Je nach Variante)
* Versionskontrolle

Die Ausbildung der beteiligten Personen, muss genügend hoch sein. Es wird aber keine Zertifizierung oder Diplome erwartet. Von den betroffenen Personen wird erwartet, dass sie selbständig Lösungswege finden und das Ziel produktiv erarbeiten.

**Dienstleistungen**

Es ist nicht vorgesehen, Dienstleistung in Anspruch zu nehmen. Wenn die Software produktiv eingesetzt wird und gehostet werden soll, muss die bei einem Hoster beantragt werden. Dies ist aber ein optionaler Teil des Betriebs und gehört somit nicht mehr zu diesem Projekt.

## Wirtschaftlichkeit

Die Projektkosten von der Wissensdatenbank belaufen sich auf null Franken. Die Zeit die wir benötigen, können wir im Unterricht verrechnen. Einen kleinen Teil werden wir zu Hause erarbeiten.

Der Nutzen hängt davon ab, ob man die Wissensdatenbank auch zweckmässig gebrauchen wird. Dies ist dann ein grosser Vorteil gegenüber den Notizen auf Blättern, da man hier das Wissen zentral und geordnet abgespeichert hat und sich das erarbeitete Wissen mit anderen teilen kann.

# Bewertung der Varianten (Tabelle)

| **Kriterium** | **Variante 1** | | **Variante 2** | | **Variante 3** | |
| --- | --- | --- | --- | --- | --- | --- |
| Abdeckung der Anforderungen | ++ |  | ++ |  | ++ |  |
| Einträge können einfach erfasst werden | + | Nicht von überall | ++ |  | ++ |  |
| Verfügbarkeit der Daten | + | Der User trägt die Verantwortung | ++ |  | ++ |  |
| Realisierbarkeit | ++ |  | ++ |  | ++ |  |
| Betrieb | ++ |  | + | Es muss einem externen Dienstleister vertraut werden | ++ | Der User kann sich zwischen Dienst und selberhosten entscheiden |
| Gesamterscheinung | ++ |  | + | Nur per Browser nutzbar | ++ |  |
| **Gesamtbeurteilung** | **+** |  | **+** |  | **++** | **Klar die flexibelste Variante** |

Bewertungen:  
++ erfüllt Kriterium optimal  
+ erfüllt Kriterium gut  
o Vor- und Nachteile halten sich die Waage  
- erfüllt Kriterium schlecht  
-- erfüllt Kriterium gar nicht

# Konsequenzen

**Abhängigkeiten**

Vom Projekt WDB sind keine anderen Projekte abhängig.

**Konsequenzen bei Nichtrealisierung**

Wenn das Projekt scheitert, muss jeder für sich eine eigene Lösung finden (analog oder digital).

**Risiken bei Nichtrealisierung**

Durch die Schwächen der IST-Situation kann die Dokumentation des Schulstoffes ungenügend ausfallen. Dadurch kann die IPA entscheidend schwieriger werden.

**Konsequenzen bei verspäteter Realisierung**

Je nach Verspätung müssen wir Notizen aufbewahren und nach der Realisierung nachtragen.

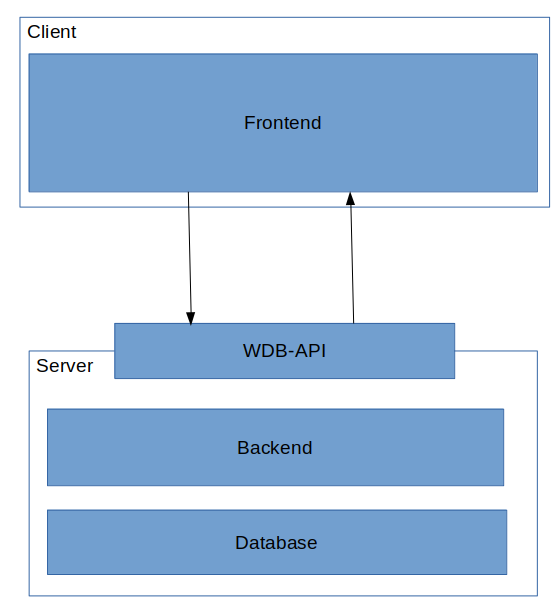
**Risiken bei verspäteter Realisierung**

Keine. Wir haben noch genug Zeit bis zur IPA.

# Lösungsbeschreibung und Empfehlung

Wir haben uns für die Variante mit einem Thin-Client und einer API entschieden. Dabei wird der Fokus in erster Linie auf Erweiterbarkeit und Funktionalität gelegt. Zudem ist das Frontend einfach austauschbar oder anpassbar. Zudem kann jeder Benutzer theoretisch sein eigenes Frontend entwickeln. Es wird ein minimales Frontend, welches die Funktionalität abdeckt, erstellt. Das Start-Frontend wird auf Basis eines Webclients erstellt. Der Server soll über eine API sämtliche wichtigen Funktionen zur Verfügung stellen.

Das Frontend wird über ein JavaScript-Framework realisiert. Auf dem Server soll ein Spring-Boot Programm laufen. Diese Trennung bietet uns die Möglichkeit, das Frontend nach spezifischen Bedürfnissen anzupassen. Spring-Boot bietet die Erstellung von APIs Out-of-the-Box an und nimmt uns dort viel Arbeit ab. Da wir nur vier Wochen für die Realisierung haben, kommt uns dies sehr entgegen. Zudem können wir dadurch bekannte Best-Practices einsetzen. Dies macht unser Programm brauchbarer und erweiterbarer.



Das Projektteam WDB empfiehlt dem Auftraggeber die Variante 3 (6.2.3), aufgrund der obigen Ausführungen, anzunehmen und zu genehmigen.

# Planung

Meilensteine und Termine

| Meilensteine | Geplant |
| --- | --- |
| Projektfreigabe | KW34 |
| Freigabe Konzept | KW38 |
| Freigabe Realisierung | KW45 |
| Freigabe Einführung | KW49 |
| Abschluss | KW03 |

Details siehe Dokument Projektplan.

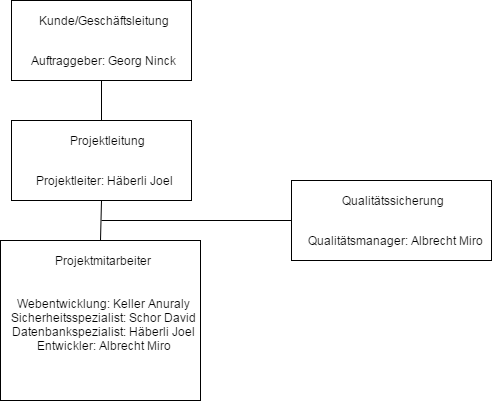
# Projektführung

## Zusammenfassung

Die Rollen sind in drei grosse Aufgabenbereiche unterteilt: Projektleitung, Qualitätssicherung und Projektmitarbeiter. So sind alle Aufgaben gut verteilbar und die Zuständigkeit für die Aufgaben innerhalb des Projektes verteilt. Berichte werden wöchentlich und mündlich an den Auftraggeber ausgestellt. Für die Qualitätssicherung ist der QM verantwortlich. Damit wir einen Überblick über unsere Resultate haben, setzten wir GitHub-Repos als Versions-Verwaltung ein. Zudem besteht eine vom Auftraggeber vorgegebene Namenskonvention für offizielle Dokumente. Die Software wird nach modernen Testmethoden und Ansätzen geprüft und entworfen.

## Projektorganisation

Die Rollen und die Namen der zugehörigen Mitarbeiter werden dem Projektinitialisierungsantrag entnommen.



## Projektberichte

Die Berichterstattung erfolgt wöchentlich. Am Ende der Doppellektion wird der Auftraggeber (Georg Ninck) mündlich über den Projektfortschritt informiert. Zudem wird im Projektplan laufend der Stand der Dinge und die aufgetauchten Probleme dokumentiert. Für die Berichterstattung ist der Projektleiter (Joel Häberli) verantwortlich. Bei Verhinderung ist er dazu verpflichtet, die Berichterstattung zu delegieren.

## Kommunikation / Projektmarketing

Die Kommunikation gegen aussen ist in der Verantwortung des Projektleiters. Die Stakeholder werden am wöchentlichen Abschlussmeeting am Ende der Doppellektion über den neusten Stand der Dinge informiert.

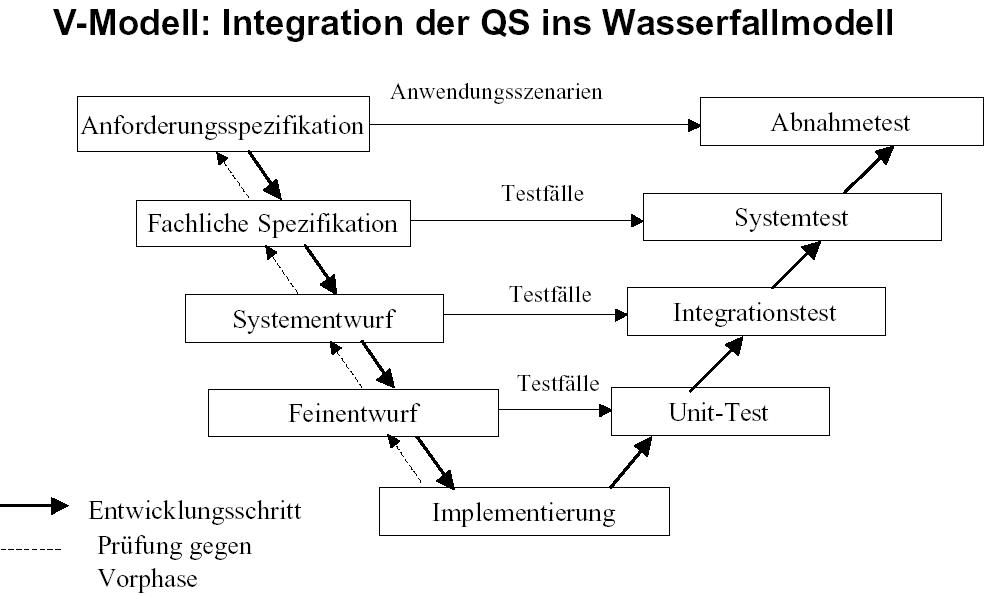
Beim Projektmarketing ist es wichtig zu betonen, dass es sich nicht um ein Lexikon wie beispielsweise Wikipedia handelt, sondern um eine persönliche Seite, auf welcher man seine Erfahrung oder Erfahrungen zentral festhalten kann.

## Qualitätssicherung

### Vorgehen zur Qualitätssicherung

Die Qualitätssicherung läuft parallel zum Projekt und mindestens jedes Bewertungsrelevante Dokument wird vor dem Abgabetermin durch den Qualitätsmanager (QM: Miro Albrecht) geprüft und allfällige Mängel an die zuständige Person zur Korrektur zurückgewiesen. Danach wird das Prozedere wiederholt. Darum ist es wichtig die Qualität laufend zu überprüfen, Mängel früh zu erkennen und zu korrigieren.

Die Codeteile werden laufend getestet und korrigiert, so dass kein „Korrekturstau“ entsteht.

**

### Qualitätsziele

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Qualitätsmerkmal** | **Messbares Qualitätsziel** | **besondere QS-Massnahmen,  um das Q.-Ziel zu erreichen** *(über die im Prüfplan vorgegebenen Massnahmen hinausgehend)* |
|  | **AUS BENUTZERSICHT** |  |  |
|  | Funktionserfüllung | Die Anwendertests sind erfolgreich | Frühes Testen der Funktionen (JUnit, TDD) |
|  | Effizienz | Die Antwortzeit des Servers entspricht den Vorgaben | Performance fördernde Massnahmen |
|  | Zuverlässigkeit | Der Server ist auch bei vielen Nutzern erreichbar | Lasttest |
|  | Benutzbarkeit | Die Anwendung ist intuitiv gestaltet | User-Experience Tests, modernes Frontend |
|  | Sicherheit | Der Datenverkehr ist verschlüsselt, heikle Daten wie Passwörter werden verschlüsselt gespeichert. | https Protokoll verwenden anstatt http, sichere Algorithmen verwenden. |
|  |  |  |  |
|  | **AUS ENTWICKLERSICHT** |  |  |
|  | Erweiterbarkeit | Neue Funktionen können einfach eingebaut werden, ohne vorher den ganzen Code um zu schreiben | Verwenden bekannter Software-Muster wie zum Beispiel MVC oder Factory-Pattern |
|  | Wartbarkeit | Der Code ist leicht konfigurierbar | Config oder Properties Dateien |
|  | Übertragbarkeit | Die eingesetzte Technologie ist modern und Plattformunabhängig | Plattformunabhängige Technologie |
|  | Wiederverwendbarkeit | Code wird in kleinst-mögliche Teile unterteilt. |  |
|  |  |  |  |
|  | **PROJEKTFÜHRUNG** |  |  |
|  | Kommunikation unter den Beteiligten | Eine Plattform über welche kommuniziert werden kann | Telegram Gruppe, Etherpad und Trello |
|  | Abstimmung mit Parallelprojekten | Wir sind ein unabhängiges Projekt und haben deshalb keine Parallelen zu anderen. Zudem ist es unser erstes Projekt. Deshalb keine Parallelen |  |
|  | Termineinhaltung | Die für die Bewertung relevanten Abgabetermine sind einzuhalten. Wenn man andere Abgabetermine versäumt ist dies dem Team sobald als möglich mitzuteilen | Kommunikation über Trello |
|  | Projektdokumentation | Die Projektdokumentation wird mit Hermes geführt. Im Projektplan wird zudem wöchentlich eine Reflexion der verrichteten Arbeit eingetragen. |  |

### Initialer Prüfplan

Übersicht über die durchzuführenden Qualitätssicherungsmassnahmen und deren Organisation (phasenweise):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Prüfobjekt** | **Termin** | **Prüfer** | **Prüfmethode** (z.B. Review,  Black-box-Test) | **Prüfkriterien, Testfälle o.ä.** (Verweis, ggf. Link) |
| Projektinitialisierungsauftrag | 23.08.2017 | Lehrperson | Review | - |
| Studie | 13.09.2017 | Lehrperson | Review | - |
| Projektplan | 13.09.2017 | Lehrperson | Review | - |
| Konzeptbericht | 01.11.2017 | Lehrperson | Review | - |
| Projektergebnis: z.B.: erstellte SW, Installation, Netzwerk­konfiguration | 29.11.2017 | QM, Projektmitarbeiter | Blackbox, Review | Software-Qualitätsmerkmale ISO 25010 |
| Realisierungs­bericht | 29.11.2017 | Lehrperson | Review,  Tests | - |
| Ergebnisbibliothek (Ablage, Directory o.ä.) |  | Lehrperson | Review | - |
| Einführungsbericht | 13.12.2017  Achtung: Ferien | Lehrperson | Review | - |
| Schlussbericht | 17.01.2017 | Lehrperson | Review | - |

### Prüfmethoden

**Review**

Der Qualitätsmanager oder eine dazu bestimmte Person prüft das Werkstück auf Unregelmässigkeiten und meldet Fehler oder korrigiert sie.

**Blackbox-Test**

Beim Blackbox-Testing sind nur Werte, die dem Werkstück übergeben werden, bekannt und welche Resultate aus diesen Werten geliefert werden sollten.

So kann dann festgestellt werden, ob eine bestimmte Funktion richtig funktioniert. Bei diesen Tests werden Grenzwerte aus der Grenzwertanalyse eingesetzt.

**White-box-Test**

Das Whitebox-Testing ist das Gegenstück zum Blackbox-Testing, also man kennt die Werte der Parameter nicht, aber man kennt den Code. Diese Methode dient zur Prüfung der Pfadabdeckung. Also dass jede Schleife berechtigt, ist und es mindestens eine Möglichkeit gibt, diese Schleife zu erreichen.

**JUnit**

Beim JUnit-Test werden einzelne Funktionen innerhalb einer Klasse auf ihre Konsistenz getestet. Die Erstellung dieser Tests ist Auftrag der Entwickler

**TDD (Test-/Entwicklungsansatz)**

TDD heisst « Test Driven Development » und heisst, dass man zuerst die Unit-Tests schreibt und erst anschliessend den zugehörigen Code. So wird überflüssiger Code verhindert. Zudem steigt die Testbarkeit der Software.

### Prüfspezifikationen

|  |  |
| --- | --- |
| **Nr.** | **Prüfkriterium** |
| 1 | Werden CleanCode Regeln eingehalten? |
| 2 | Werden bekannte Software-Pattern verwendet? |
| 3 | Liefern die einzelnen Funktionen die richtigen Resultate? |
| 4 | Sind die Methoden kleinstmöglich? |
| 5 | Werden Properties oder Configurations verwendet? |

### Checklisten für die Prüfung der Projektdokumente

Die Checkliste für die Prüfung der Dokumente findet man im Dokument „306.05 Bewertungsdossier“. Dieses liegt auf dem Share: /sh-classes/inf2015\_5d\_306/K\_0\_Initialisierungsantrag.

### Testfalltabellen

In der Konzeptphase wird damit begonnen, Testfälle zu sammeln, soweit sie bei der Erarbeitung der Systemanforderungen und der Systemarchitektur bereits erkannt werden können.

In der Realisierungsphase werden die Testfalltabellen in den Realisierungsbericht übernommen und dort weiter detailliert.

Die Testfalltabellen sollen mindestens folgende Information enthalten:

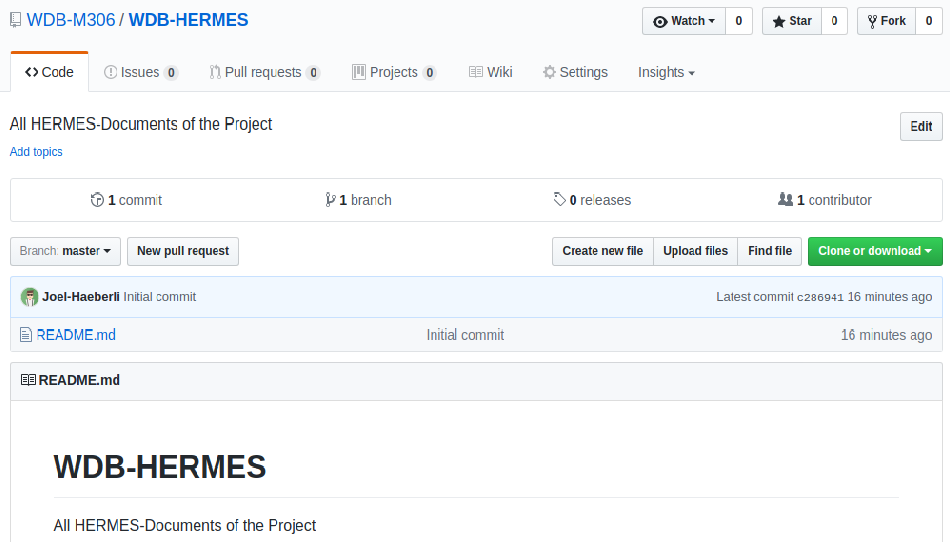
* Nummer
* Kurzbezeichnung
* Allenfalls abgedeckten Anwendungsfall
* Erwartetes Ergebnis
* Bemerkungen

## Konfigurationsmanagement

### Konfigurationsidentifikation

|  |  |
| --- | --- |
| **Ergebnis** bei SW-Bestandteilen: gemäss Systemarchitektur | **Ablage** |
| Software | https://github.com/WDB-M306/WDB |
| Hermes-Dokumente | https://github.com/WDB-M306/WDB-HERMES |

### Ablagestruktur

  
Die Ablagestruktur ist sehr einfach. Alle Dokumente im Rahmen der Bewertung werden in oben aufgelisteten GitHub-Repository abgelegt, ohne weitere Strukturierung.

### Namenskonventionen

Die Dokumente, welche man dem Auftraggeber übergeben muss verfolgen eine Namenskonvention, welche vom Auftraggeber vorgegeben wurde:

[PHASENNUMMER]\_[DOKUMENTNUMMER]\_[DOKUMENTBEZEICHNUNG]\_WDB.docx

Begriffe in []-Klammern durch betreffende Werte ersetzen.

Legende:

* Phasennummer: Initialisierung = 1, Einführung = 4
* Dokumentnummer: siehe „306.3 Lieferobjekte und Terminliste“

Dokumentbezeichnung: Studie, Projektplan, usw.

# Projektantrag

Hiermit beantragen wir, dass das Projekt **WDB** mit der Lösungsvariante **Variante 3** für die Phase Konzept freigegeben wird.

12. September, Bern, Anuraly Keller, David Schor, Joel Häberli und Miro Albrecht