

Table of Contents

Projekt Episko	2
Kickoff Meeting	3
Planning Meeting	4
Entwicklungsplan Meeting	8
Meetingtitel	9
Abnahme Entwicklungsplan	11
Definition Use-Cases	12
Abnahme Anforderungsanalyse	13
Inkmente	15
Initialisierung Inkrement 0	16
Projektskizze	17
Entwicklungsplan	20
Anforderungsanalyse	27
Initialisierung Inkrement 1	55
Initialisierung Inkrement 2	56
Dokumententitel	57

Projekt Episko

Toller Name

Wieso

Weshalb

Warum

Kickoff Meeting

Übersicht

Projekt: Projekt Episkos *Iteration:* 0 *Arbeitspaket:* Projektinitialisierung **Datum, Ort:** 01.10.2024, DHBW Friedrichshafen **Teilnehmer:** Simon Blum, Ben Oeckl, Paul Stoeckle, Max Rodler, Sascha Wanninger **Moderation:** Sascha Wanninger **Diskussionspunkte:** \

- Vorstellung der Projektskizze
- Feedback des Auftraggebers

Ergebnisse: \

- "Go" für das Projekt wurde gegeben
- Feedback zur Spezifizierung der Stakeholder

Aktionen:

Aktion	Verantwortlich	Deadline
Feedback umsetzen	Max Roder	07.10.2024
Abgabe der Projektskizze	Ben Oeckl	11.10.2024
Link zu Dokument reparieren	Simon Blum	21.11.2024

Relevante Dokumente

- Projektskizze ([Projektskizze](#))

Planning Meeting

Übersicht

Projekt: Projekt Episkos *Iteration:* 0 *Arbeitspaket:* Projektinitialisierung **Datum, Ort:** 01.10.2024, DHBW Friedrichshafen **Teilnehmer:** Simon Blum, Ben Oeckl, Paul Stoeckle, Max Rodler **Moderation:** Max Rodler **Diskussionspunkte:** \

- Was ist unser Ziel?
- Wie erreichen wir dieses?

Ergebnisse: \

- Erstellung der Projektskizze

Aktionen:

Aktion	Verantwortlich	Deadline
Meeting mit Auftraggeber	Alle	01.10.2024

Projekt Ziele

Ziel	Was soll erreicht werden?
Stakeholder	Zielgruppe, Entwickler, Abnehmer
Auswirkungen auf Stakeholder	Einfachere Verwaltung von Projekten
Randbedingungen	Zeitraumen (6 Monate), Vorgaben für das Projekt
Abhängigkeiten	Hauptziel - Keine Abhängigkeiten
Sonstiges	Klare Struktur und Dokumentation

Rahmenbedingungen

Risiken

Go - Checklist

- ☐ Sind die Ziele klar und eindeutig?
 - Ja
- ☐ Sind die Ziele messbar?
 - Messbar auf Basis von Feedback
 - Aufwand manuell vs mit Anwendung
- ☐ Bedeuten die Ziele einen klaren Vorteil für den Kunden/Anwender?
 - Ja, Ziel ist es das Verwalten und die Übersicht von Projekten signifikant zu vereinfachen
- ☐ Kann man die Ziele in der gegebenen Zeit und mit dem gegebenen Budget erreichen?
 - Ja

- [] Gibt es Risiken mit hoher Wahrscheinlichkeit, die es unmöglich machen das Projekt erfolgreich durchzuführen?
 - Nein, wir sind flexibel und zuversichtlich alles überwinden zu können
- [] Sind alle Stakeholder bereit mitzuarbeiten?
 - ?
- [] Gibt es weitere Untersuchungen, die vor dem Start durchgeführt werden müssen?
 - Findung von Technologien etc.
 - Marktanalyse - gibt es schon ähnliche Produkte?

Goal concept

- Project mangement system

Das Ziel des Projektes ist es eine Anwendung zu erstellen, welche genutzt werden kann um Programmierprojekte zu verwalten. Hierfür soll eine konsolenbasierte und eine graphische Anwendung existieren. Das System soll über eine standartisierte Manifestdatei ermöglicht werden. Folgende Funktionen sollen ermöglicht werden:

- Übersicht über vorhandene Projekte
 - Name, Pfad...
- Sortierung durch Kategorieren/Labels
- Kreation und verwaltung von Projekten

Zukunft


Zusätzlich kann hierbei erweitert werden mit:

- Integration Git/Github
 - Status
- Statistiken

- Öffnen in favorisierter IDE
- Möglichkeiten der Fernverwaltung

Systemgrenzen

- Interaktion mit Metadaten der Projekte
- Keine Interaktion mit Projekten selbst (paketmanagement, deployment, etc)

 [!Note] Das Design der Anwendung soll flexibel genug sein um diesen Grenzen in zukünftigen Aufwänden erweitern zu können und so mehr Funktionalität einzubinden.

Todo:

- [] Team orga
- [] Projektname
- [] Technologien

Entwicklungsplan Meeting

Übersicht

Projekt: Projekt Episkos *Iteration:* 0 *Arbeitspaket:* Projektinitialisierung **Datum, Ort:** 11.10.2024, DHBW Friedrichshafen **Teilnehmer:** Simon Blum, Ben Oeckl, Paul Stoeckle, Max Rodler **Moderation:** Max Rodler **Diskussionspunkte:** \

- Genauer Aufbau der Vorgehensweise
- Aufgabenverteilung
- Struktur UML-Diagramms/Vorgehensmodell

Ergebnisse: \

- Inhaltliche Fertigstellung des Entwicklungsplans

Aktionen:

Aktion	Verantwortlich	Deadline
Diagramme formalisieren	Simon Blum	11.10.2024
Formales dokument erstellen	Max Rodler	18.10.2024
Link zu Dokument reparieren	Simon Blum	21.11.2024

Relevante Dokumente

- Entwicklungsplan ([Entwicklungsplan](#))

Meetingtitel

08-11-2024-Use-Cases

Übersicht

Projekt: Projekt Episkos *Iteration:* 0 *Arbeitspaket:* Anforderungsanalyse **Datum, Ort:** 08.11.2024 , DHBW Friedrichshafen **Teilnehmer:** Simon Blum, Ben Oeckl, Paul Stoeckle, Max Rodler **Moderation:** Max Rodler **Diskussionspunkte:** \

- Anforderungen und Use-Cases definieren

Ergebnisse: \

- Basisinformationen
- Prozessdaten
- Funktionalitäten
- Angrenzende Systeme

Aktionen:

Aktion	Verantwortlich	Deadline
Anforderungsanalyse erstellen	Alle	15.11.2024

Basisinformationen

- Metadatenverwaltung
 - Neues Projekt anlegen
 - Recursives auffinden von Manifestdateien
 - Einlesen von Manifestdateien

- Cachen von Manifesten
- Metadaten bearbeiten
- Projekt löschen

Prozessdaten

- Kein vorhanden

Funktionalität

- Detaillierte Übersicht, Analysen
- Suchen, Filtern

Angrenzende Systeme

- Keine, ggf. in Zukunft git

Sonstiges

- Auf Folien auch allgemeinerer Ablauf zusammengefasst

Relevante Dokumente

Abnahme Entwicklungsplan

Übersicht

Projekt: Projekt Episko *Iteration:* 0 *Arbeitspaket:* Projektinitialisierung **Datum, Ort:** 13.11.2024, DHBW-Friedrichshafen **Teilnehmer:** Simon Blum, Ben Oeckl, Paul Stoeckle, Max Rodler, Sascha Wanninger **Moderation:** Sascha Wanninger **Diskussionspunkte:**

- Vorstellung Entwicklungsplan
- Abnahme Entwicklungsplan

Ergebnisse:

- Begriff Anforderungsanalyse ändern
- Entwicklungsplan abgenommen

Aktionen:

Aktion	Verantwortlich	Deadline
Begriff Anforderungsanalyse im Inkrement ändern	Simon Blum	15.11.2024
Link zu Dokument reparieren	Simon Blum	21.11.2024

Relevante Dokumente

- Entwicklungsplan ([Entwicklungsplan](#))

Definition Use-Cases

Übersicht

Projekt: Projekt Episko *Iteration:* 0 *Arbeitspaket:* Projektvorbereitung **Datum, Ort:** 13.11.2024, DHBW-Friedrichshafen **Teilnehmer:** Simon Blum, Ben Oeckl, Paul Stoeckle, Max Rodler **Moderation:** Simon Blum **Diskussionspunkte:**

- Gedanken zu Use-Cases machen
- Use-Case Schablone ausfüllen
- zugehörige funktionale Anforderungen finden
- nichtfunktionale Anforderungen definieren

Ergebnisse:

- Use-Cases 1.1 - 3.4 definiert
- zugehörige funktionale Anforderungen formuliert
- nichtfunktionale Anforderungen formuliert

Aktionen:

Aktion	Verantwortlich	Deadline
Diagramme erstellen	Simon Blum	15.11.24
Anforderungen umformulieren nach Rupp	Paul Stöckle	15.11.24
Zuordnung zu den use-Cases	Ben Oeckl	15.11.24

Relevante Dokumente

Abnahme Anforderungsanalyse

Übersicht

Projekt: Projekt Episko *Iteration:* 0 *Arbeitspaket:* Projektinitialisierung **Datum, Ort:** 15.11.2024, DHBW-Friedrichshafen **Teilnehmer:** Simon Blum, Ben Oeckl, Paul Stoeckle, Max Rodler, Sascha Wanninger **Moderation:** Sascha Wanninger **Diskussionspunkte:**

- Vorstellung & Abnahme Anforderungsanalyse

Ergebnisse:

- Schritte abstrahieren (loslösen Von GUI/CLI (teilweise gleiche Schritte))
- Nicht der USER serialisiert (Soll kein Use-Case sein)
- Checksum erstellen ist auch kein Use-Case
- U3.4 Titel fehlt
- Metadaten laden auch kein Use-Case
- F0.1.1 nicht funktional
- Funktionale Requirements umformulieren (Nutzer weniger im Fokus)

Aktionen:

Aktion	Verantwortlich	Deadline
Funktionale Requirements umformulieren	Paul Stöckle	19.11.2024
Use-Cases überarbeiten (abstrahieren)	Simon Blum	19.11.2024
Link zu Dokument reparieren	Simon Blum	21.11.2024

Relevante Dokumente

- Anforderungsanalyse ([Anforderungsanalyse](#))

Inkremente

N r.	Titel	Begin	Deadli ne	Tats. Ende	Arbeits pakete	Verantwo rtlicher	Initialisierung
0	Initialisi erung	01.10.2 024	18.10.2 024	-	-	Max Rodl er	(Link)[00/Initialis ierung0.md]
1	-	-	-	-	-	-	(Link)[01/Initialisi erung1.md]
2	-	-	-	-	-	-	(Link)[02/Initialis ierung2.md]

Arbeitspakete

Übersicht über Arbeitspakete ist in den Ordnern der Dokumente.

Initialisierung Inkrement 0

Übersicht

Projekt: Projekt Episkos *Iteration:* *Arbeitspaket:* Autor: Datum: Zuletzt geändert:
von: am: Version: 1 Prüfer: Letzte Freigabe: durch: am: \

Changelog

Datum	Verfasser	Kurzbeschreibung
XX.XX.XXXX	?????????	Initiales Erstellen und Verfassen

Distribution List

- Simon Blum simon21.blum@gmail.com (<mailto:simon21.blum@gmail.com>)
- Ben Oeckl ben@oeckl.com (<mailto:ben@oeckl.com>)
- Maximilian Rodler maximilianreinerrodler@gmail.com
(<mailto:maximilianreinerrodler@gmail.com>)
- Paul Stöckle paul.stoeckle@t-online.de (<mailto:paul.stoeckle@t-online.de>)

Zeitplan

Arbeitspakete

Projektskizze

Übersicht

Projekt: Projekt Episko *Iteration:* 0 *Arbeitspaket:* Projektinitialisierung **Autor:** Simon Blum **Datum:** 01.10.2024 **Zuletzt geändert:** *von:* Ben Oeckl *am:* 11.10.2024 **Version:** 4
Letzte Freigabe: *durch:* Simon Blum *am:* 11.10.2024

Changelog

Datum	Verfasser	Kurzbeschreibung
01.10.2024	Simon Blum	Initiales Erstellen und Verfassen
01.10.2024	Paul Stöckle	Hinzufügen von Formalitäten
01.10.2024	Maximilian Rodler	Anpassung an Feedback nach "Go/No-Go" Meeting
07.10.2024	Simon Blum	Aktualisierung von Formalitäten
11.10.2024	Ben Oeckl	Finalisierung zur Abgabe

Distribution List

- Simon Blum simon21.blum@gmail.com (<mailto:simon21.blum@gmail.com>)
- Ben Oeckl ben@oeckl.com (<mailto:ben@oeckl.com>)
- Maximilian Rodler maximilianreinerrodler@gmail.com
(<mailto:maximilianreinerrodler@gmail.com>)
- Paul Stöckle paul.stoeckle@t-online.de (<mailto:paul.stoeckle@t-online.de>)

Scope of Work / Projektziele

- Verwaltung und Übersicht von Programmierprojekten
- Erstellung eines "Manifeststandards" für Metadaten eines Projektes
- Erstellung einer Kommandozeilenanwendung zum initialisieren/betrachten dieser Metadaten
- Erstellung einer Graphischen Anwendung zur Verwaltung und Übersicht

Systemgrenzen

- Interaktion mit Metadaten der Projekte
- Keine Interaktion mit Projekten selbst (Paketmanagement, Deployment, etc.)



[!Note] Das Design der Anwendung soll flexibel genug sein um diesen Grenzen in zukünftigen Aufwänden erweitern zu können und so mehr Funktionalität einzubinden.

Risiken

- Vorerst keine identifiziert

Stakeholder

- Projektinterne Entwickler
- Auftraggeber (Dozent)
- Kunden:
 - Professionelle Entwickler
 - Hobbyentwickler
 - Studentische Entwickler

Randbedingungen

- Zeitraum 6 Monate
- Vorgaben zur Projektorganisation

Entwicklungsplan

Übersicht

Projekt: Projekt Episko *Iteration:* 0 *Arbeitspaket:* Projektentwicklungsplan **Autor:** Maximilian Rodler **Datum:** 11.10.2024 **Zuletzt geändert:** von: Maximilian Rodler am: 11.10.2024 **Version:** 1 **Letzte Freigabe:** durch: Simon Blum am: 11.10.2024

Changelog

Datum	Verfasser	Kurzbeschreibung
01.10.2024	Maximilian Rodler	Vorgehensmodell entwickeln und ausarbeiten
12.10.2024	Simon Blum	Diagramme hinzufügen

Distribution List

- Simon Blum simon21.blum@gmail.com (<mailto:simon21.blum@gmail.com>)
- Ben Oeckl ben@oeckl.com (<mailto:ben@oeckl.com>)
- Maximilian Rodler maximilianreinerrodler@gmail.com (<mailto:maximilianreinerrodler@gmail.com>)
- Paul Stöckle paul.stoeckle@t-online.de (<mailto:paul.stoeckle@t-online.de>)

Vorgehensmodell

Gearbeitet wird primär Iterativ-Inkrementell:

- Zu Beginn des Projektes werden Inkremente (Meilensteine) geplant.
- Dieses Inkrement kann ein oder mehrere Arbeitspakete beinhalten.

Diese Iterativ-Inkrementelle Arbeitsweise beinhaltet agile Elemente:

- Es gibt wöchentliche Meetings zur Absprache.
- Hier können die Ziele/der Zeitraum eines Inkrements angepasst werden, falls nötig.
- Es können parallel mehrere Arbeitspakete in einem Inkrement durchgeführt werden.

Eine genauere Übersicht kann unter " Diagramme (["Diagramme" in "Entwicklungsplan"](#)) " gefunden werden.

Teamzusammensetzung

- Simon Blum, TIT23
- Paul Stöckle, TIT23
- Maximilian Rodler, TIT23
- Ben Oeckl, TIT23

Rollen

Projektmanager – Maximilian Rodler Head of Development – Paul Stöckle Head of Quality & Operations – Simon Blum Head of Testing & Integrations – Ben Oeckl

Verantwortlichkeiten und Aufgaben

Aufgabe	Verantwortliche
Protokollierung	Maximilian Rodler
Erstellung und Verwaltung von Arbeitspaketen/Meilensteine	Maximilian Rodler, Paul Stöckle
Einhaltung und Planung von Deadlines	Maximilian Rodler
Sicherung von Codequalität	Simon Blum
Sicherung von Dokumentenqualität	Simon Blum
Erstellung und Verwaltung von CI/CD Pipelines	Ben Oeckl, Simon Blum
Übersichtlichkeit und Struktur der Organisation	Ben Oeckl, Simon Blum
Testen der Funktionalität	Ben Oeckl
Koordinierung und Verantwortlichkeit für Arbeiten am Source Code	Paul Stöckle

Dokumente

Im Laufe des Prozesses werden diverse Dokumente erstellt.

Meetings

Zu jedem Meeting wird ein Dokument erstellt welches die folgenden Informationen beinhaltet:

- Datum, Ort
- Teilnehmer
- Moderation

- Diskussionspunkte
- Ergebnisse
- Folgeaktionen
- *Optional:* Hinweis auf relevante Dokumente
- *Optional:* Notizen

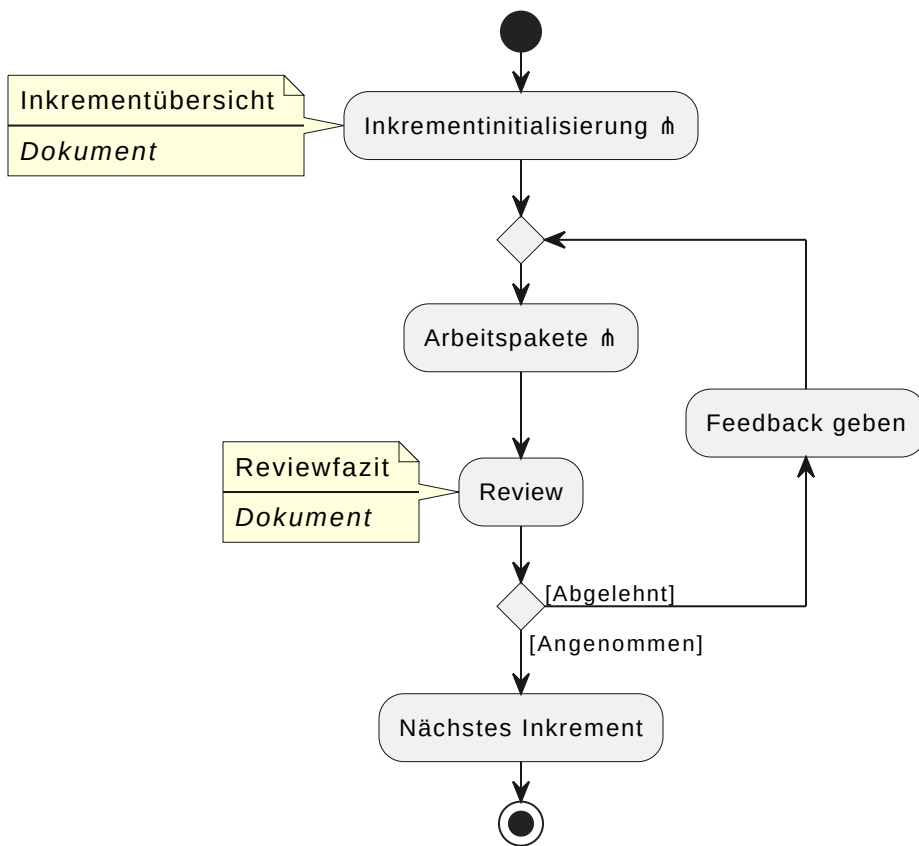
Inkremente

Im Rahmen eines Inkrementes werden folgende Dokumente erstellt:

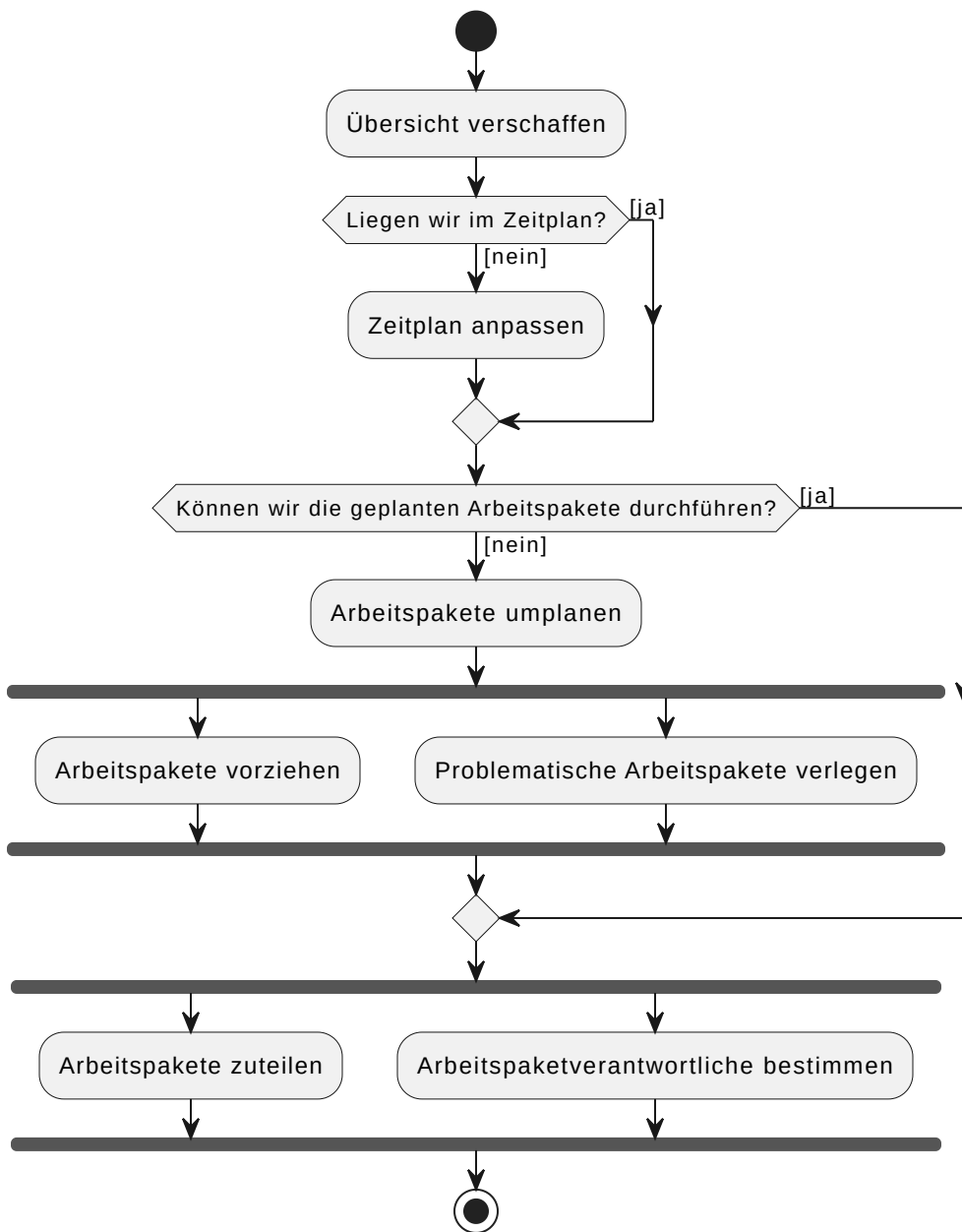
- Inkrementübersicht
 - Beinhaltet Arbeitspakete und Verantwortliche des Projektes
- Reviewfazit
 - Dieses wird bei mehreren Reviews erweitert
- Dokumente der/des Arbeitspaket/es:
 - Anforderungskatalog
 - Designpaper
 - Entwicklerdokumentation
 - Abschlussreport
 - *Wird das Arbeitspaket in einer Iteration überarbeitet, werden diese Dokumente ergänzt!*

Diagramme

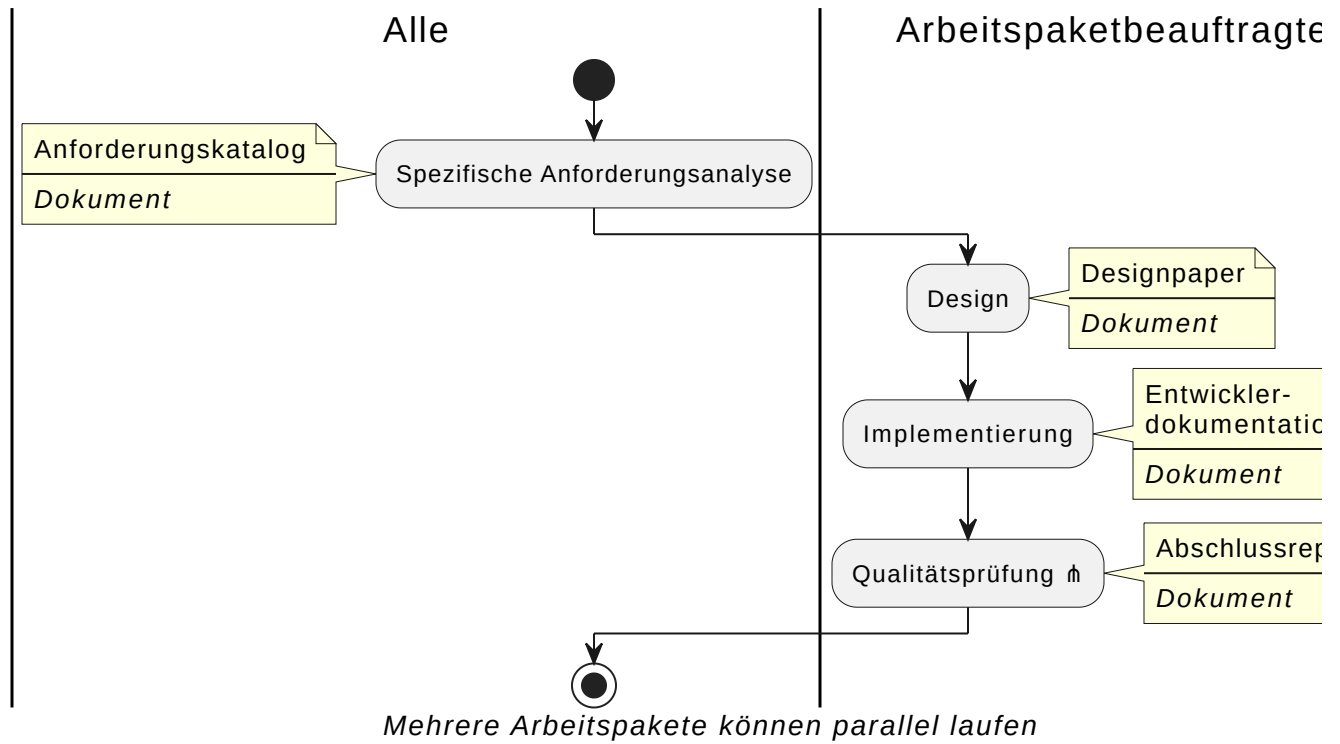
Vorgehensmodell



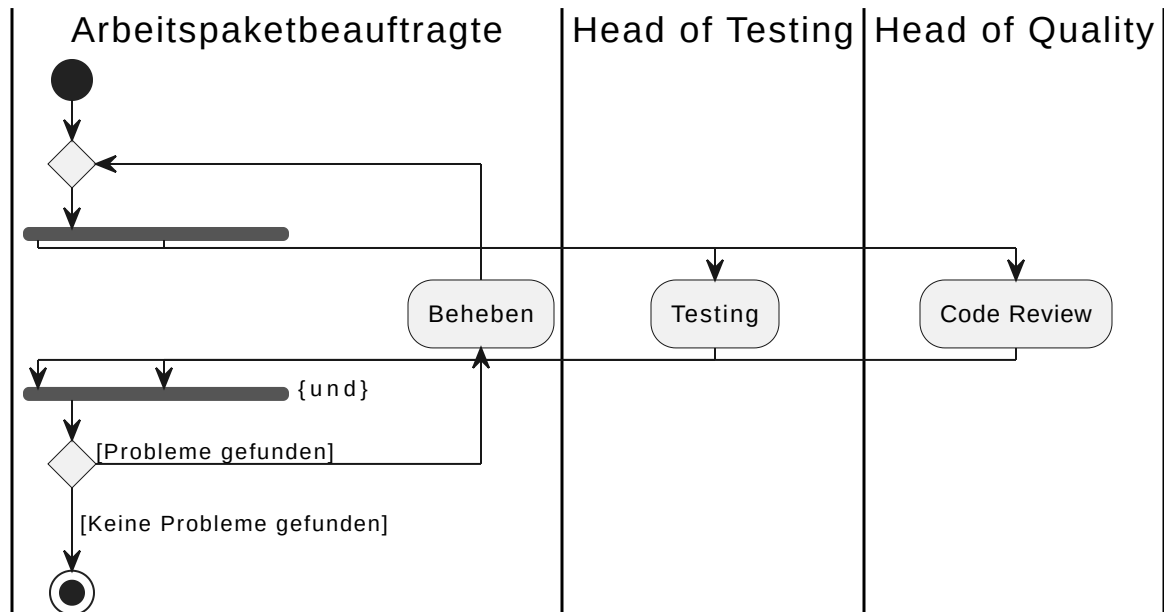
Initialisierung



Inkrement



Qualitätsprüfung



Anforderungsanalyse

Dokumentdaten

Übersicht

Projekt: Projekt Episkos *Iteration:* 0 *Arbeitspaket:* Anforderungsanalyse **Autor:** Simon Blum **Datum:** 13.11.2024 **Zuletzt geändert:** von: Paul Stöckle am: 15.11.2024 **Version:** 6
Prüfer: Paul Stöckle **Letzte Freigabe:** durch: Max Rodler am: 15.11.2024

Changelog

Datum	Verfasser	Kurzbeschreibung
13.11.2024	Simon Blum	Initialer Meeting-Mitschrieb
13.11.2024	Ben Oeckl	Ergänzung von Requirements in Use Cases
13.11.2024	Paul Stöckle	Überarbeitung von Requirements
15.11.2024	Paul Stöckle	Hinzufügen des Headers
15.11.2024	Max Rodler	Fehlerbehebung
21.11.2024	Simon Blum	Aktualisierung von UseCases und Requirements

Distribution List

- Simon Blum simon21.blum@gmail.com (<mailto:simon21.blum@gmail.com>)
- Ben Oeckl ben@oeckl.com (<mailto:ben@oeckl.com>)
- Maximilian Rodler maximilianreinerrodler@gmail.com (<mailto:maximilianreinerrodler@gmail.com>)
- Paul Stöckle paul.stoeckle@t-online.de (<mailto:paul.stoeckle@t-online.de>)

UseCases

Erläuterung Pakete

Die UseCases sind in 3 Pakete aufgeteilt.

Paket 1 - Create, Read, Update, Delete

Bei den UseCases in Paket 1 geht es primäre um die atomare manipulation von Daten.

Paket 2 - Manifestinteraktion

Bei den UseCases in Paket 2 geht es vorallem um die Interaktionen mit dem lokalen Dateisystem und sich dort befindende Manifestdateien

Paket 3 - Funktionalitäten

Paket 3 umfasst UseCases welche erweiterte Funktionalitäten des Systems darstellen.

UC1.1 Anwendung starten

	Iteration	
Id	1	UC1.1
Paket	2	1
Autor	1	
Version	1	4
Kurzbeschreibung	1	Der Nutzer kann die Anwendung starten
Beteiligte Akteure	1	Nutzer
Fachverantwortlicher	1	
Referenzen	2	
Vorbedingungen	2	Die Anwendung ist auf einem kompatiblen System installiert
Nachbedingungen	2	Die Anwendung ist gestartet und nutzungsbereit
Typischer Ablauf	2	s. Ablaufdiagramm
Alternative Abläufe	3	-
Kritikalität	3	0
Verknüpfungen	2	UC2.1, U2.2

Funktionale Anforderungen	4	FA1.1.1, FA1.1.2, FA2.1.5, FA2.2.5
Nicht-funktionale Anforderungen	4	NA2

UC1.1 UseCase Diagramm

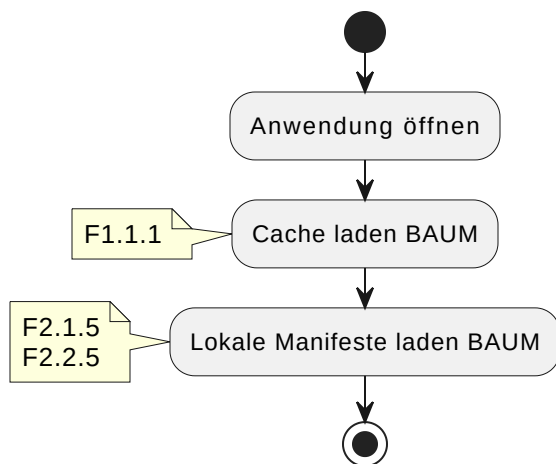
Dot Executable: /opt/local/bin/dot
 Dot executable does not exist
 Cannot find Graphviz. You should try

```
@startuml
testdot
@enduml
```

or

```
java -jar plantuml.jar -testdot
```

UC1.1 Ablaufdiagramm



UC1.2 Metadaten anlegen

Metadaten anlegen	Iteration	
Id	1	UC1.2
Paket	2	P1
Autor	1	
Version	1	5
Kurzbeschreibung	1	Der Nutzer kann mithilfe der Anwendung oder manuelle eine Manifestdatei mit Metadaten erstellen
Beteiligte Akteure	1	Nutzer
Fachverantwortlicher	1	
Referenzen	2	Dateiformat Doku
Vorbedingungen	2	Es muss ein Verzeichnis für das Projekt existieren , in dem sich keine andere Manifestdatei befindet
Nachbedingungen	2	Es existiert eine Manifestdatei in dem gewählten Ordner. Wurde das Projekt über die Anwendung erstellt, wurden die Daten in der Datenbank gecached.
Typischer Ablauf	2	s. Ablaufdiagramm

Alternative Abläufe	3	s. Ablaufdiagramm
Kritikalität	3	0
Verknüpfungen	2	
Funktionale Anforderungen	4	FA1.2.1, FA1.2.2, FA1.2.3
Nicht-funktionale Anforderungen	4	

UC1.2 UseCase Diagramm

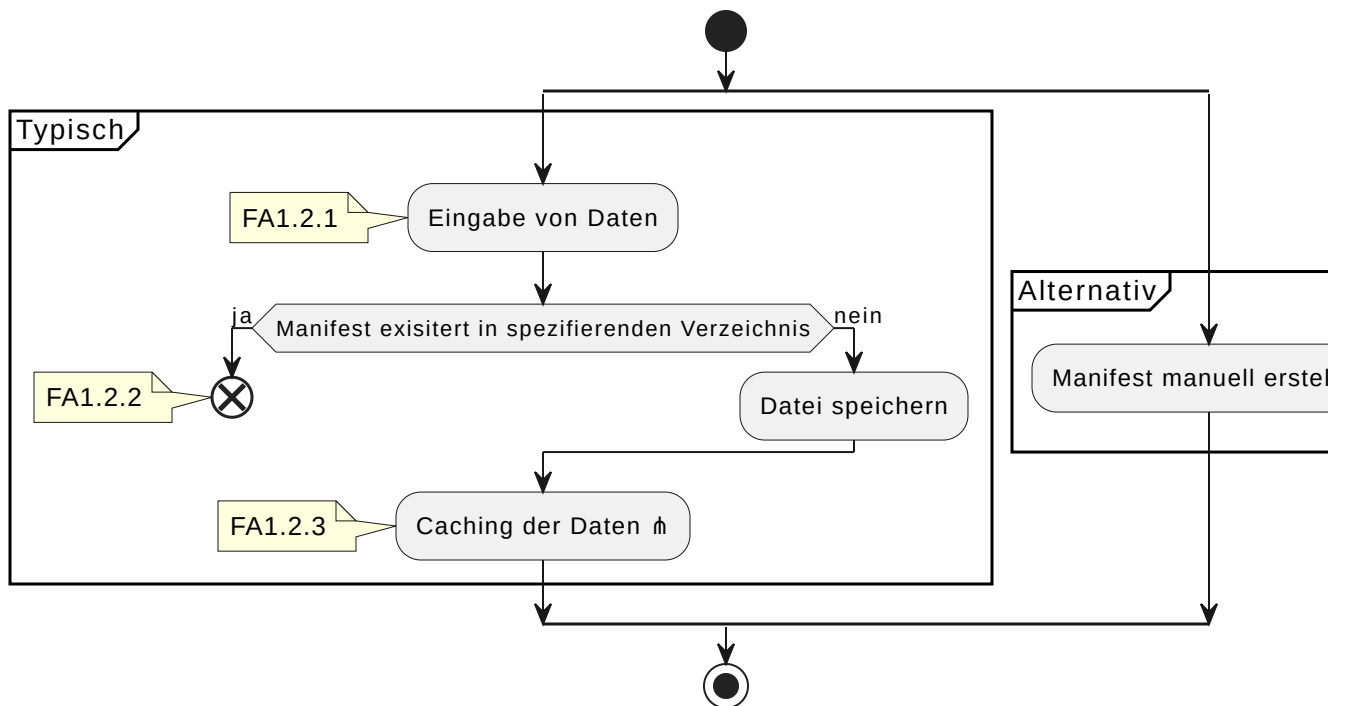
Dot Executable: /opt/local/bin/dot
Dot executable does not exist
Cannot find Graphviz. You should try

```
@startuml
testdot
@enduml
```

or

```
java -jar plantuml.jar -testdot
```

UC1.2 Ablaufdiagramme



UC1.3 Metadaten bearbeiten

Schritt	Iteration	
Id	1	UC1.3
Paket	2	P1
Autor	1	
Version	1	5
Kurzbeschreibung	1	Der Nutzer kann die Metadaten eines Projektes über die Anwendung oder manuell in der Datei bearbeiten.
Beteiligte Akteure	1	Nutzer
Fachverantwortlicher	1	
Referenzen	2	Dateiformat Doku
Vorbedingungen	2	Es muss eine Manifestdatei existieren die bearbeitet werden kann.
Nachbedingungen	2	Die angepasste Manifestdatei wird gespeichert. Wurde die Datei über die Anwendung geändert, werden die Änderungen gecached.
Typischer Ablauf	2	s. Ablaufdiagramm

Alternative Abläufe	3	s. Ablaufdiagramm
Kritikalität	3	1
Verknüpfungen	2	
Funktionale Anforderungen	4	FA1.3.1, FA1.3.2, FA1.3.3
Nicht-funktionale Anforderungen	4	

UC1.3 UseCase Diagramm

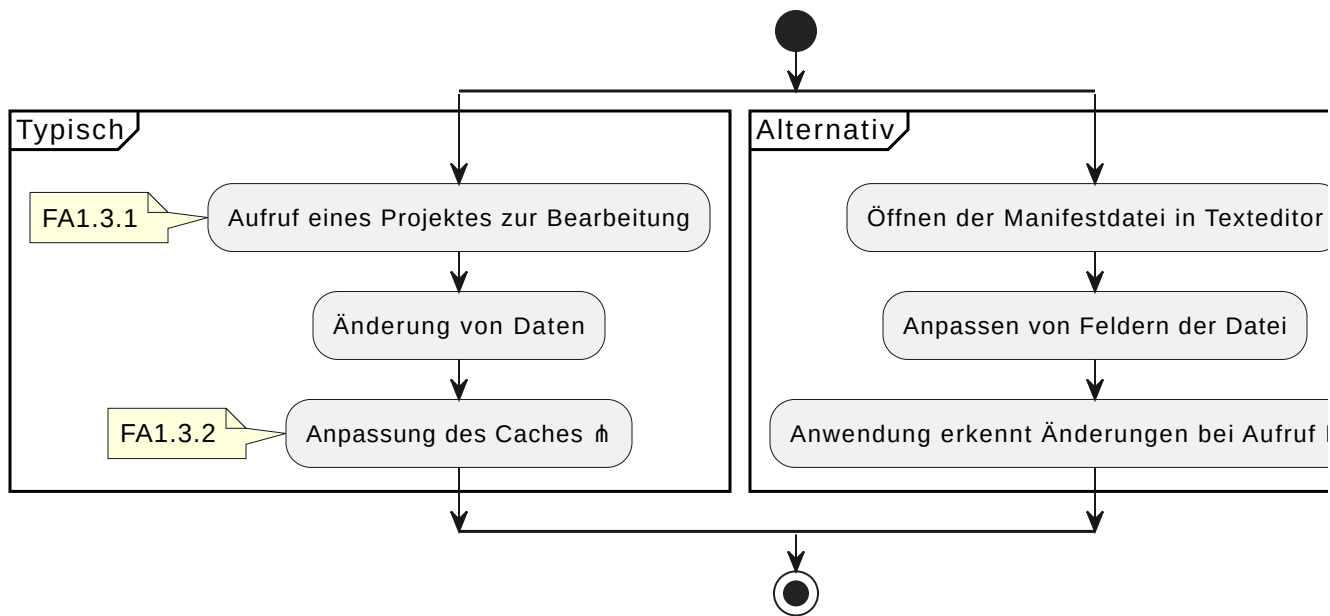
Dot Executable: /opt/local/bin/dot
 Dot executable does not exist
 Cannot find Graphviz. You should try

```
@startuml
testdot
@enduml
```

or

```
java -jar plantuml.jar -testdot
```

UC1.3 Ablaufdiagramme



UC1.4 Metadaten löschen

Schritt	Iteration	
Id	1	UC1.4
Paket	2	P1
Autor	1	
Version	1	5
Kurzbeschreibung	1	Der Nutzer kann die Metadaten für ein Projekt löschen
Beteiligte Akteure	1	Nutzer
Fachverantwortlicher	1	
Referenzen	2	
Vorbedingungen	2	Es existiert eine valide Manifestdatei die gelöscht werden kann
Nachbedingungen	2	Es existiert keine Manifestdatei mehr. Bei manueller Löschung wird der Cache im nachhinein, beim nächsten Starten der Anwendung aktualisiert.
Typischer Ablauf	2	s. Ablaufdiagramm

Alternative Abläufe	3	s. Ablaufdiagramm
Kritikalität	3	1
Verknüpfungen	2	Beim löschen über die Anwendung UseCase 3.1
Funktionale Anforderungen	4	FA1.4.1, FA1.4.2, FA1.4.3
Nicht-funktionale Anforderungen	4	

UC1.4 UseCase Diagramm

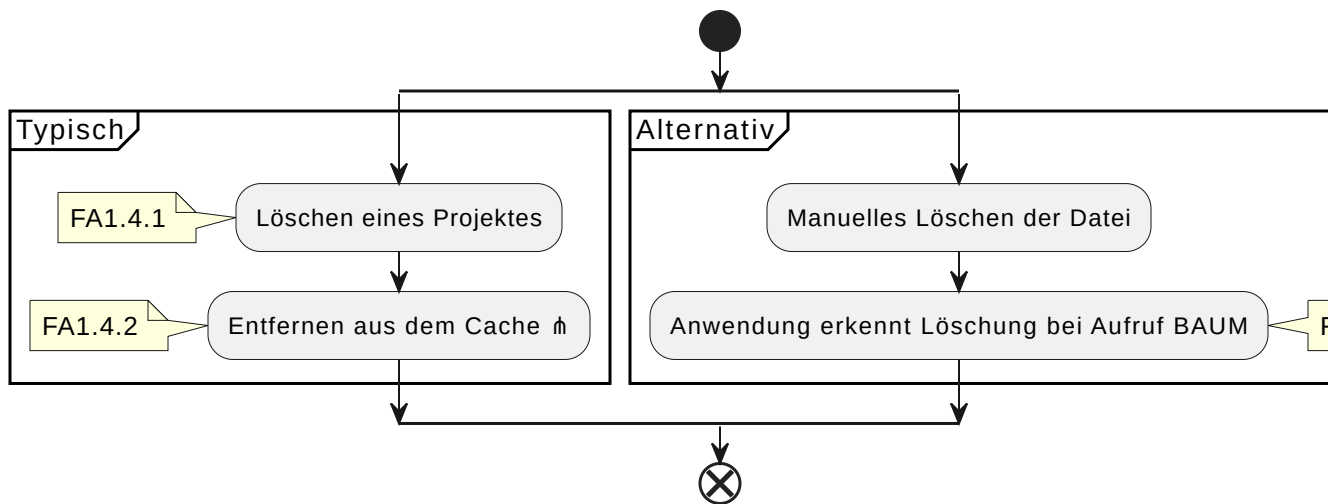
Dot Executable: /opt/local/bin/dot
Dot executable does not exist
Cannot find Graphviz. You should try

```
@startuml
testdot
@enduml
```

or

```
java -jar plantuml.jar -testdot
```

UC1.4 Ablaufdiagramme



UC2.1 Datei angeben

	Iteration	
Id	1	UC2.1
Paket	2	2
Autor	1	
Version	1	4
Kurzbeschreibung	1	Der Nutzer kann den Pfad zu einer Manifestdatei angeben, welche dann deserialisiert wird. Der Pfad der Datei kann gespeichert werden und beim nächsten Starten der Anwendung erneut deserialisiert werden.
Beteiligte Akteure	1	Nutzer
Fachverantwortlicher	1	
Referenzen	2	-
Vorbedingungen	2	Es existiert eine valide Manifestdatei welche der Nutzer angeben kann.
Nachbedingungen	2	Die Datei wurde deserialisiert und die Daten können weiterverarbeitet werden.
Typischer Ablauf	2	s. Ablaufdiagramm

Alternative Abläufe	3	-
Kritikalität	3	0
Verknüpfungen	2	U1.1
Funktionale Anforderungen	4	FA2.1.1, FA2.1.2, FA2.1.3, FA2.1.4, FA2.1.5
Nicht-funktionale Anforderungen	4	

UC2.1 UseCase Diagramm

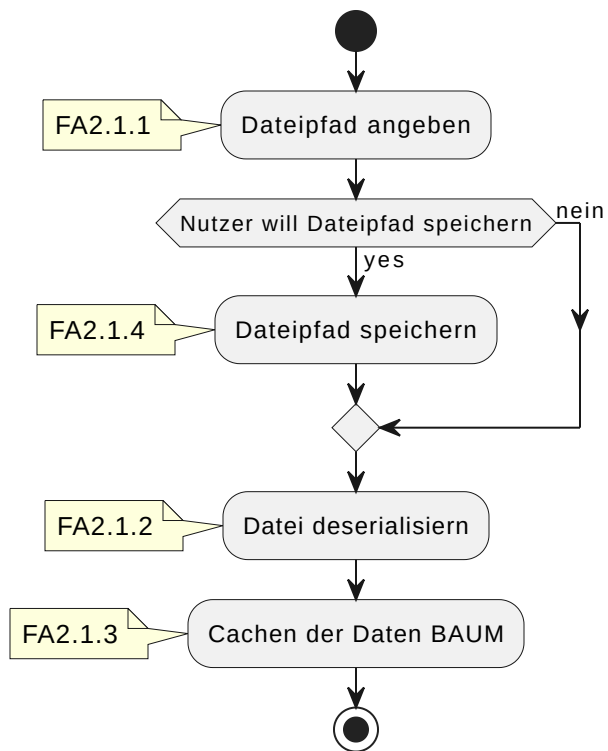
Dot Executable: /opt/local/bin/dot
Dot executable does not exist
Cannot find Graphviz. You should try

```
@startuml
testdot
@enduml
```

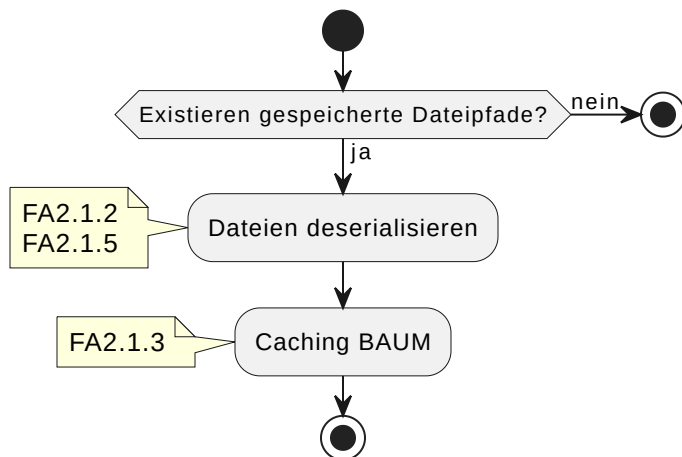
or

```
java -jar plantuml.jar -testdot
```

UC2.1 Ablaufdiagramm



Ablauf bei Start der Anwendung



UC2.2 Verzeichnis angeben

	Iteration	
Id	1	UC2.2
Paket	2	2
Autor	1	
Version	1	4
Kurzbeschreibung	1	Der Nutzer kann einen Pfad angeben, welcher rekursiv nach Manifesten durchsucht wird. Angegebene Pfade können gespeichert werden und beim nächsten Ausführen der Anwendung wieder durchsucht werden.
Beteiligte Akteure	1	Nutzer
Fachverantwortlicher	1	
Referenzen	2	
Vorbedingungen	2	Es existiert ein Verzeichnis welches der Nutzer angeben kann.
Nachbedingungen	2	Wenn in dem Verzeichnis Manifeste liegen, wurden diese deserialisiert.
Typischer	2	s. Ablaufdiagramm

Ablauf		
Alternative Abläufe	3	s. Ablaufdiagramm
Kritikalität	3	
Verknüpfungen	2	U1.1
Funktionale Anforderungen	4	FA2.2.1, FA2.2.2, FA2.2.3, FA2.2.4, FA2.2.5
Nicht-funktionale Anforderungen	4	

UC2.2 UseCase Diagramm

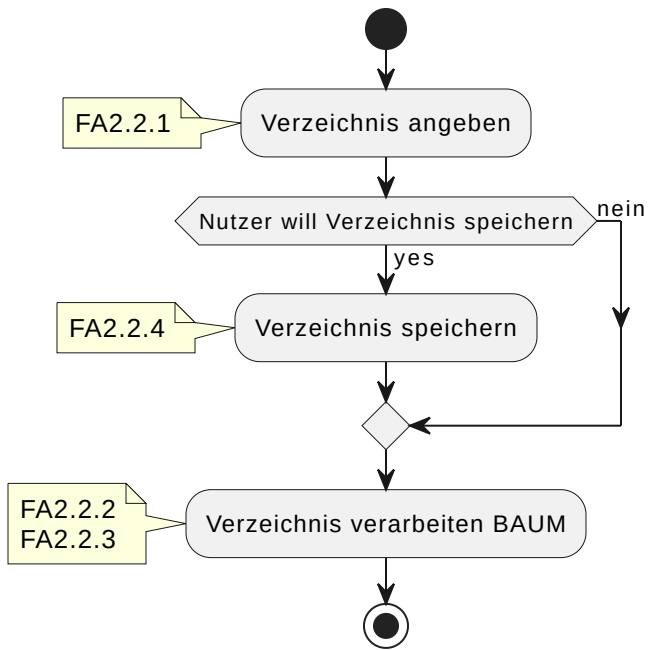
Dot Executable: /opt/local/bin/dot
Dot executable does not exist
Cannot find Graphviz. You should try

```
@startuml
testdot
@enduml
```

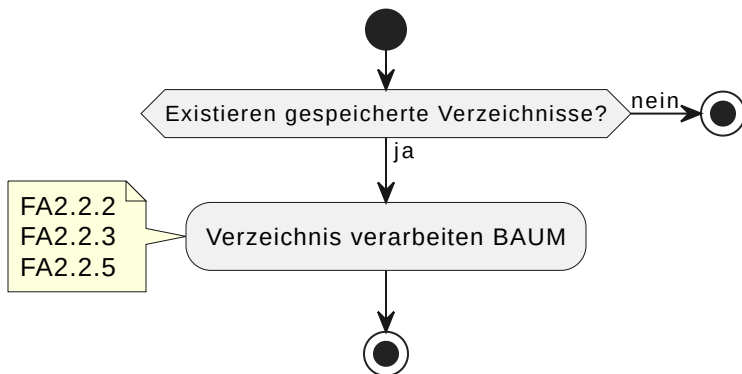
or

```
java -jar plantuml.jar -testdot
```

UC2.2 Ablaufdiagramm



Ablauf bei Start der Anwendung



UC3.1 Projekte suchen

Schritt	Iteration	
Id	1	UC3.1
Paket	2	P3
Autor	1	
Version	1	6
Kurzbeschreibung	1	Der Nutzer kann seine Projekte nach verschiedenen Eigenschaften durchsuchen
Beteiligte Akteure	1	Nutzer
Fachverantwortlicher	1	
Referenzen	2	
Vorbedingungen	2	Die Anwendung ist gestartet und gecachete und lokale Daten wurden geladen.
Nachbedingungen	2	Dem Nutzer werden die Projekte angezeigt, die den gegebenen Eigenschaften entsprechen
Typischer Ablauf	2	s. Ablaufdiagramm
Alternative Abläufe	3	-
Kritikalität	3	3

Verknüpfungen	2	-
Funktionale Anforderungen	4	FA3.1.1, FA3.1.2, FA3.1.3
Nicht-funktionale Anforderungen	4	

UC3.1 UseCase Diagramm

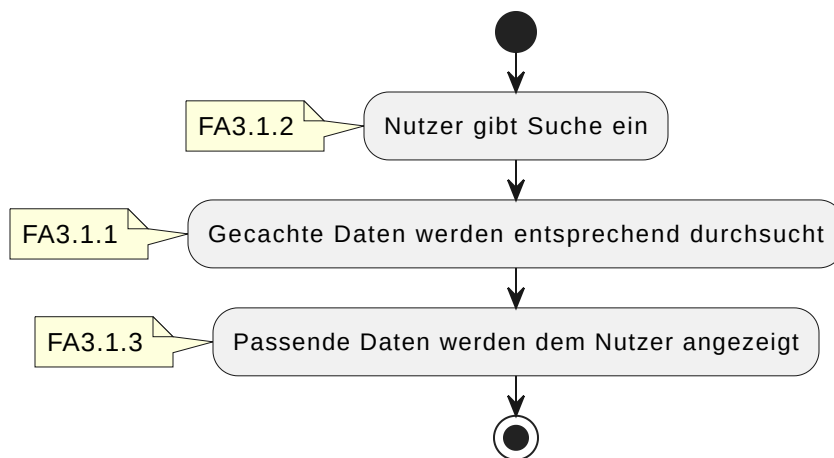
Dot Executable: /opt/local/bin/dot
 Dot executable does not exist
 Cannot find Graphviz. You should try

@startuml
 testdot
 @enduml

or

java -jar plantuml.jar -testdot

UC3.1 Ablaufdiagramm



UC3.2 Projekte filtern

Schritt	Iteration	
Id	1	UC3.2
Paket	2	P3
Autor	1	
Version	1	6
Kurzbeschreibung	1	Der Nutzer kann seine Projekte nach verschiedenen Kriterien filtern
Beteiligte Akteure	1	Nutzer
Fachverantwortliche	1	
Referenzen	2	
Vorbedingungen	2	Die Anwendung ist gestartet und gecachete und lokale Daten wurden geladen.
Nachbedingungen	2	Dem Nutzer werden die Projekte angezeigt, die den gegebenen Kriterien entsprechen
Typischer Ablauf	2	s. Ablaufdiagramm
Alternative Abläufe	3	-
Kritikalität	3	3

Verknüpfungen	2	-
Funktionale Anforderungen	4	FA3.2.1, FA3.2.2, FA3.2.3
Nicht-funktionale Anforderungen	4	

UC3.2 UseCase Diagramm

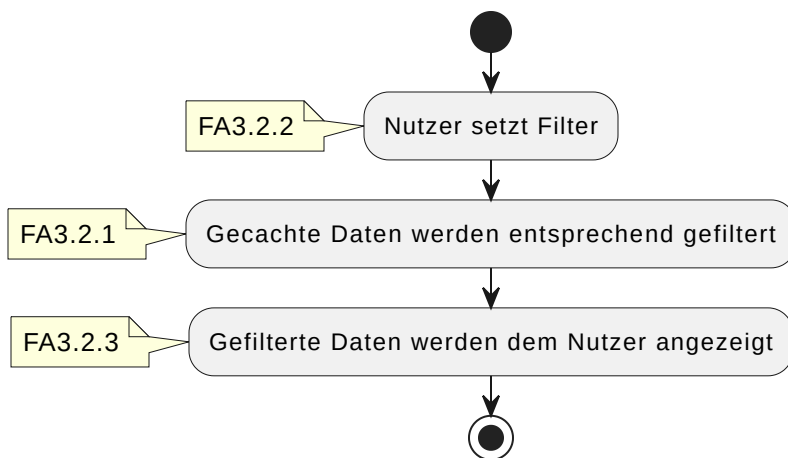
Dot Executable: /opt/local/bin/dot
 Dot executable does not exist
 Cannot find Graphviz. You should try

@startuml
 testdot
 @enduml

or

java -jar plantuml.jar -testdot

UC3.2 Ablaufdiagramm



UC3.3 Statistiken

Schritt	Iteration	
Id	1	UC3.3
Paket	2	P3
Autor	1	
Version	1	5
Kurzbeschreibung	1	Dem Nutzer werden in der graphischen Anwendung verschiedene Statistiken und Analysen zu seinen Projekten bereitgestellt
Beteiligte Akteure	1	Nutzer
Fachverantwortlicher	1	
Referenzen	2	
Vorbedingungen	2	Die Anwendung ist gestartet und gecachete und lokale Daten wurden geladen.
Nachbedingungen	2	Der Nutzer kann die bereitgestellten Statistiken in der graphischen Anwendung einsehen
Typischer Ablauf	2	s. Ablaufdiagramm
Alternative Abläufe	3	-

ufe		
Kritikalität	3	3
Verknüpfungen	2	
Funktionale Anforderungen	4	FA3.3.1, FA3.3.2
Nicht-funktionale Anforderungen	4	

UC3.3 UseCase Diagramm

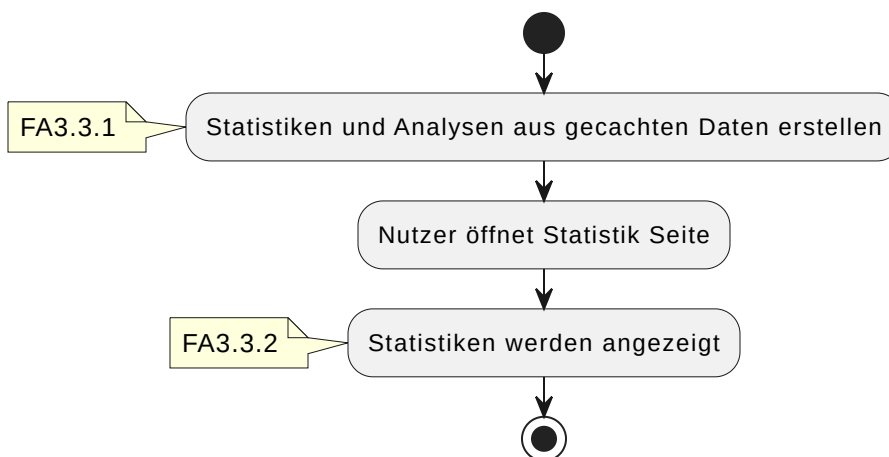
Dot Executable: /opt/local/bin/dot
 Dot executable does not exist
 Cannot find Graphviz. You should try

```
@startuml
testdot
@enduml
```

or

```
java -jar plantuml.jar -testdot
```

UC3.3 Ablaufdiagramm



Anforderungen

Funktionale Anforderungen

FA1.1.1 Beim Starten soll das System gecachte Daten laden.

FA1.1.2: Das System muss Manifestdateien aus dem lokalen Datensystem deserialisieren.

FA1.2.1 Das System muss dem Nutzer die Möglichkeit bieten eine Manifestdatei mit Metadaten zu erstellen.

FA1.2.2 Beim Erstellen muss das System prüfen ob in dem relevanten Verzeichnis bereits eine Manifestdatei existiert.

FA1.2.3 Nach Erstellen soll das System die Metadaten im Cache speichern.

FA1.3.1 Das System soll dem Nutzer die Möglichkeit bieten Metadaten anzupassen.

FA1.3.2 Bei Änderungen soll das System relevante Metadaten automatisch im Cache aktualisieren.

FA1.3.3 Bei manuellen Änderungen an der Datei muss das System diese erkennen und dementsprechende Anpassungen im Cache vornehmen.

FA1.4.1 Das System muss dem Nutzer die Möglichkeit bieten Metadaten zu löschen.

FA1.4.2 Bei Löschung soll das System die relevanten Daten automatisch aus dem Cache entfernen.

FA1.4.3 Bei manueller Löschung muss das System dies erkennen und den relevanten Eintrag aus dem Cache entfernen.

FA2.1.1 Das System muss dem Nutzer die Möglichkeit bieten den Pfad zu einer einzelnen Manifestdatei anzugeben.

FA2.1.2 Gibt der Nutzer den Pfad zu einer valide Datei ein, muss das System in der Lage sein diese zu deserialisieren.

FA2.1.3 Nach der Deserialisierung soll das System die Daten im Cache speichern.

FA2.1.4 Das System soll dem Nutzer die Möglichkeit bieten Dateipfade für zukünftiges deserialisieren zu speichern.

FA2.1.5 Wenn gespeicherte Dateipfade existieren soll, das System beim Starten diese automatisch deserialisieren.

FA2.2.1 Das System soll dem Nutzer die Möglichkeit bieten ein Verzeichnis anzugeben, welches rekursiv nach Manifestdateien durchsucht wird.

FA2.2.2 Wenn in diesem Verzeichnis Manifeste existieren soll das System diese serialisieren.

FA2.2.3 Wenn das System ein Manifest aus einem Verzeichnis serialisiert hat, soll es bei Abweichungen den Cache aktualisieren.

FA2.2.4 Das System soll dem Nutzer die Möglichkeit bieten Verzeichnisse für zukünftiges durchsuchen zu speichern.

FA2.2.5 Wenn gespeicherte Verzeichnisse existieren, soll das System beim Starten diese automatisch durchsuchen.

FA3.1.1 Das System soll gecachte Metadaten auf verschiedene Eigenschaften durchsuchen können.

FA3.1.2 Das System soll dem Nutzer die Möglichkeit bieten die Suche anzupassen.

FA3.1.3 Das System soll dem Nutzer die Möglichkeit bieten auf Suchergebnisse zugreifen zu können.

FA3.2.1 Das System soll gecachte Metadaten nach verschiedenen Kriterien filtern können.

FA3.2.2 Das System soll dem Nutzer die Möglichkeit bieten den Filter anzupassen.

FA3.2.3 Das System soll dem Nutzer die Möglichkeit bieten auf den gefilterten Datensatz zugreifen zu können.

FA3.3.1 Das System soll Statistiken aus gecachten Metadaten erstellen können.

FA3.3.2 Das System soll dem Nutzer die Möglichkeit bieten auf diese Statistiken zugreifen zu können.

Nicht funktionale Anforderungen

NA1: Die Manifestdateien müssen von Menschen, als auch von Maschinen lesbar sein.

NA1.1: Die in der Manifestdatei zu findenden Metadaten sollen dem Nutzer nützliche Informationen über das dazugehörige Projekt bieten.

NA2: Die Anwendung soll schnellst möglich dem Nutzer nach dem Start zur Bedienung bereit stehen.

NA3: Die Anwendung soll möglichst responsiv und nutzerfreundlich sein.

NA4: Die Anwendung muss in den Betriebssystem Microsoft Windows 10, Microsoft Windows 11, und Linux funktionieren.

NA4.1: Für Linux sollen Pakete in den Formaten für die Distributionen/Paketsystem Debian/Ubuntu (apt), Arch (pacman) und Nix (nixpkgs).

NA5: Die Anwendung soll in Rust und Typescript verfasst sein.

NA5.1: Für die Anwendung sollen die Frameworks "Tauri v2.0+" für das Backend und "SvelteKit v2.8+" für das Frontend genutzt werden.

NA6: Der "Cache" der Anwendung soll als persistenter Cache mithilfe einer SQLite Datenbank implementiert werden.

NA6.1: Bei der Implementierung der Datenbank muss darauf geachtet werden, dass diese vor SQL-Injektionen ausreichend gesichert ist.

NA7: Für relevante Subsysteme müssen Unittests verfasst werden.

NA8: Die Anwendung muss für die Prozessorarchitektur x86_64 ausgelegt sein.

NA9: Die Anwendung muss Barrierefrei konstruiert werden um bspw. die Nutzung von Screenreadern zu erlauben.



Da die Anwendung mithilfe von Webtechnologien gebaut werden soll mehr Infos bspw. hier (<https://www.aktion-mensch.de/inklusion/barrierefreiheit/barrierefreie-website>)

NA10: Die Anwendung soll zunächst mit der Oberflächensprache Deutsch oder Englisch gebaut werden.

NA10.1: Texte in der Oberfläche sollen so eingebaut um zukünftig die Implementierung neuer Sprachen einfach zu gestalten.

NA11: Das Projekt muss bis zum Ende der Theoriephase im Quartal 1 im Jahr 2025 abgeschlossen sein. Ein exaktes Datum hierfür folgt.

NA12: Alle Meetings müssen in Meetingprotokollen festgehalten werden.

NA13: Es muss eine Entwicklerdokumentation angefertigt werden.

Initialisierung Inkrement 1

Übersicht

Projekt: Projekt Episkos *Iteration:* *Arbeitspaket:* Autor: Datum: Zuletzt geändert:
von: am: Version: 1 Prüfer: Letzte Freigabe: durch: am: \

Changelog

Datum	Verfasser	Kurzbeschreibung
XX.XX.XXXX	?????????	Initiales Erstellen und Verfassen

Distribution List

- Simon Blum simon21.blum@gmail.com (<mailto:simon21.blum@gmail.com>)
- Ben Oeckl ben@oeckl.com (<mailto:ben@oeckl.com>)
- Maximilian Rodler maximilianreinerrodler@gmail.com
(<mailto:maximilianreinerrodler@gmail.com>)
- Paul Stöckle paul.stoeckle@t-online.de (<mailto:paul.stoeckle@t-online.de>)

Zeitplan

Arbeitspakete

Initialisierung Inkrement 2

Übersicht

Projekt: Projekt Episkos *Iteration:* *Arbeitspaket:* Autor: Datum: Zuletzt geändert:
von: am: Version: 1 Prüfer: Letzte Freigabe: durch: am: \

Changelog

Datum	Verfasser	Kurzbeschreibung
XX.XX.XXXX	?????????	Initiales Erstellen und Verfassen

Distribution List

- Simon Blum simon21.blum@gmail.com (<mailto:simon21.blum@gmail.com>)
- Ben Oeckl ben@oeckl.com (<mailto:ben@oeckl.com>)
- Maximilian Rodler maximilianreinerrodler@gmail.com
(<mailto:maximilianreinerrodler@gmail.com>)
- Paul Stöckle paul.stoeckle@t-online.de (<mailto:paul.stoeckle@t-online.de>)

Zeitplan

Arbeitspakete

Dokumententitel

Übersicht

Projekt: Projekt Episkos *Iteration:* 0 *Arbeitspaket:* - **Autor:** Simon Blum **Datum:** 11.11.2024 **Zuletzt geändert:** - *von:* *am:* **Version:** 1 **Prüfer:** Ben Oeckl **Letzte Freigabe:** *durch:* *am:* \

Changelog

Datum	Verfasser	Kurzbeschreibung
11.11.2024	Simon Blum	Initiales Erstellen und Verfassen

Distribution List

- Simon Blum simon21.blum@gmail.com (<mailto:simon21.blum@gmail.com>)
- Ben Oeckl ben@oeckl.com (<mailto:ben@oeckl.com>)
- Maximilian Rodler maximilianreinerrodler@gmail.com (<mailto:maximilianreinerrodler@gmail.com>)
- Paul Stöckle paul.stoeckle@t-online.de (<mailto:paul.stoeckle@t-online.de>)

Begriff	Erklärung
Metadaten	Repräsentierung von Stammdaten eines Projektes im Zuge der Software. Eine genaue Definierung welche Daten hier beinhaltet sind erfolgt im Laufe des ersten Inkrements.
Projekt	Softwareprojekt des Nutzers welches durch Metadaten beschrieben werden kann.
Manifest(datei)	Repräsentierung von Metadaten in einer Datei. Die genaue Struktur wird im Laufe des ersten Inkrements festgelegt
Inkrement	Zeitraum in dem festgelegte Arbeitspakete bearbeitet werden. Abschluss dieser kann auch als Meilenstein bezeichnet werden
Meilenstein	Abschluss eines Inkrementes
Kritikalität	Im Rahmen des Projektes kann die Kritikalität verschiedener ??? folgendermaßen angegeben werden: 0 - Absolut unabdingbar, 1 - Sehr wichtig, 2 - Normal 3 - Unwichtig
Serialisieren	
Deserialisieren	
Gecachte Date	Metadaten welche in einer Datenbank zwischengespeichert sind um das wiederholte deserialisieren von Manifestdateien zu vermeiden

Lokale Daten	Metadaten welche sich in Manifestdateien befinden und nicht gecached sind
Cache	Der Begriff Cache deutet im Rahmen des Projektes auf einen persistenten Cache in Form einer Datenbank hin. Diese ist nur ein "Cache" da die Manifeste die "single source of thruth" sind.