

# Projektkonzeption und -realisierung

*W3WI\_106.2 und W3WI\_107*

Dozent: Prof. Dr. Michael Eichberg

Kontakt: [michael.eichberg@dhbw-mannheim.de](mailto:michael.eichberg@dhbw-mannheim.de), Raum 149B

Version: 23SEB

# Einführung

## 1. Allgemeines

1. Teaser des Projekts
2. Vorstellung des Ablaufs der Projektkonzeption und -realisierung
3. Zusammenfinden in 3 Teams
4. Kurze Wiederholung von relevanten Themen

## 2. Start der Projektkonzeption

---

# Das Projekt

Entwicklung eines AppStores für Apps zum Deployment auf OpenStack.

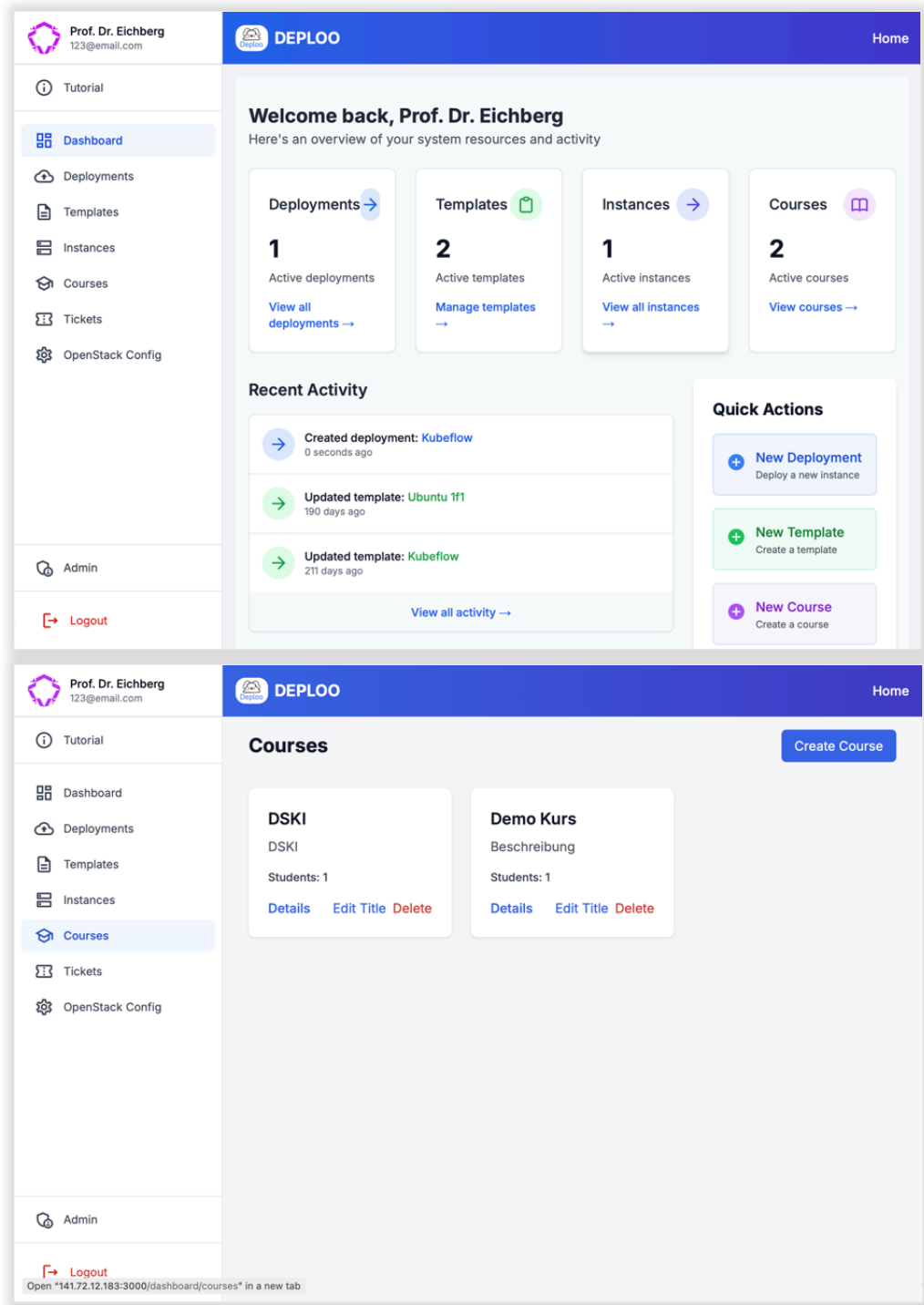
## Ziel

Dozenten können - ohne tiefergehende Kenntnisse von OpenStack etc. - Anwendungen auf der OpenStack Platform deployen, die dann durch Studierende im Rahmen von Lehrveranstaltungen genutzt werden können.

Den Dozenten soll eine einfache Administrierbarkeit der Anwendungen ermöglicht werden.

Den Administratoren ist es stets möglich Dozenten zu unterstützen.

# Erster Prototyp der Gruppe Deplooo



Prof. Dr. Eichberg  
123@email.com

Tutorial

Dashboard

Deployments

Templates

Instances

Courses

Tickets

OpenStack Config

Admin

Logout

DEPLOO

Home

Templates

+ Create Template

Type: All types Visibility: All templates Sort by: Name Type Date Visibility

Docker

Public

Kubeflow

Lightweight Python IPYNB  
enviroment with preinstal...

Ian

One for one  
Working

Delete

Details

Image

Public

Ubuntu 1f1

No description

Tyler

1f1 script

Delete

Details

Prof. Dr. Eichberg  
123@email.com

Tutorial

Dashboard

Deployments

Templates

Instances

Courses

Tickets

OpenStack Config

Admin

Logout

DEPLOO

Home

OpenStack Configuration

OpenStack Configuration

clouds.yaml configured

clouds.yaml Configuration

Choose File no file selected

Delete Configuration

Help

The clouds.yaml file is required for OpenStack authentication. You can obtain this file from your OpenStack dashboard:

1. Log in to your OpenStack dashboard

2. Go to Sidebar → Identität → Applikations-Zugangsdaten

3. Click on "Applikations-Zugangsdaten erstellen"

4. Fill in the required fields

5. Download the clouds.yaml file

Open "141.72.12.183-3000/dashboard/openstack-config" in a new tab

# Beispielanwendungen

- Eine einfache VM mit NodeJS für Studierende bzw. Gruppen von Studierenden (ggf. inkl. Quotas und Porteinschränkungen)
- VMs für Pentesting Zwecke:
  - VMs die als Angriffsziele dienen
  - VMs mit Angriffswerkzeugen
  - ggf. in einem zugangsgeschützten virtuellen Netzwerk
- Ein Gitlabserver für das Verwalten von Projekten eines bestimmten Kurses
- Ein Jenkins Server für das Durchführen von Continuous Integration
- Ein Jupyter Notebook Server für die Durchführung von Übungen
- Eine vollständige Netzwerkinfrastruktur
- Eine OpenStack Umgebung für Lehrveranstaltungen im Bereich Cloud Computing
- Ein Kubernetes Cluster für das Deployment von Microservices
- Ein Kubernetes Cluster für KI Anwendungen
- Eine vorkonfigurierte Entwicklungsumgebung für die Programmierung (z.B. in Java mit Eclipse Theia)
- Projektmanagementsoftware (z. B. OpenProject)
- Ein Datenbankserver mit passenden Accounts (z. B. MySQL, PostgreSQL)
- ...

# Anforderungen bzw. Herausforderungen

## Funktionale Anforderungen an den AppStore

- Es soll sowohl vorgefertigte Anwendungen (bzw. Anwendungstemplates) geben, die einfach deployt werden können, als auch möglich sein dem App Store eigene (private) Anwendungen hinzuzufügen.
- Beim Deployment von Anwendungen werden die notwendigen Konfigurationsschritte (möglichst) automatisiert durchgeführt:
  - Anlegen der entsprechenden Anzahl an VMS
  - Anlegen von Nutzeraccounts bzw. Gruppenaccounts (ggf. basierend auf CSV Dateien, manuell oder von einem LDAP Server); alternativ Konfiguration von SSH Keys
  - Einrichten von Netzwerkkonfigurationen (ggf. inkl. Erzeugung von Zertifikaten)
  - Konfiguration von DNS.
  - ...
- Es ist dem Dozenten ggf. möglich (einzelne) Anwendungen neu zu starten bzw. neu zu konfigurieren (z. B. wenn Anwendungen nicht mehr reagieren weil zum Beispiel im Rahmen des Pentesting die ZielVMs abgestürzt sind).
- Anwendungen können in unterschiedlichen Versionen vorliegen.
- Anwendungen können einfach aktualisiert werden (z. B. können die Anwendungen in einem GIT liegen und ein Dozent bzw. Administrator kann darüber Aktualisierungen durchführen).
- Jedem Dozenten sind nur begrenzte Ressourcen zur Verfügung gestellt (z. B. nur 10 VMs). Es kann ggf. sinnvoll sein, mehrere Instanzen einer Anwendung in einer VM zu deployen.

## Nicht-funktionale Anforderungen

- Der App-Store ist eine eigenständige Webanwendung, der die öffentlichen Schnittstellen von OpenStack verwendet.  
(D.h. er ist kein Plug-in.)
- Der Technologiestack sollte überschaubar sein, damit das Projekt ggf. später weiterentwickelt/übernommen werden kann.  
(Dies ist Lizenz und Ergebnisabhängig.)

---

Jedes Team ist frei in der Wahl der Softwarelizenz. Sie können sich auch entscheiden Ihr Projekt unter keine explizite Lizenz zu stellen.

Eine weitere Übernahme des Projekts ist natürlich nur dann möglich, wenn Sie eine entsprechende Open-Source Lizenz wählen (Informationen finden Sie z.B. auf: <https://www.linux-magazin.de/ausgaben/2025/04/software-lizenzen-teil-1/> oder auf <https://www.computerweekly.com/de/definition/Softwarelizenz> bzw.

auf <https://opensource.org/>).

**Die Benotung des Projekts ist unabhängig von der gewählten Lizenz.**

- Wartbarkeit und Qualität sind primäre Ziele.

#### ※ Hinweis

Die genauen Features bzw. Anforderungen legen wir gemeinsam in den folgenden Wochen im Rahmen von *Requirements Workshops* fest.

Die oben genannten Anforderungen stellen nur den initialen, groben Rahmen dar.



# Möglicherweise relevante Technologien

Sowohl für die Entwicklung des App Stores, als auch der Anwendungen (ggf. als Inspiration):

## © Bemerkung

Abgesehen von OpenStack als Deploymentziel sind Sie frei in der Wahl der Technologien.

- Docker
- Kubernetes
- FluxCD

Flux is a tool for keeping Kubernetes clusters in sync with sources of configuration, and automating updates to configuration when there is new code to deploy.

- HELM Charts

Charts describe even the most complex apps, provide repeatable application installation, and serve as a single point of authority.

- Open Component Model

The Open Component Model (OCM) is your one-stop open-source Software Bill of Delivery (SBoD) for packaging, signing, transporting and deploying your artifacts – preserving end-to-end security, integrity and provenance.

# Benotungsgrundlage

## Projektkonzeption

Abgaben:

- **[1P]** Produktvorführung
- **[47P]** Dokumentationen und Präsentationen **[1]**
- **[2P je Person]** Selbstbezogener Reflexionsbericht

## Projektrealisierung

Abgaben:

- **[18P]** Zwischenpräsentation und aktualisierte Dokumentation (am Ende des ersten Semesters)
- **[78P]** das Produkt (am Ende des nächsten Semesters/des Projekts ist das Produkt als solches vorzuführen und abzugeben)
- **[4P]** die qualitätsgesicherte Nutzerdokumentation
- **[7P]** die qualitätsgesicherte Entwicklerdokumentation
- **[4P]** ein Tutorialvideo
- **[5P]** Entwicklervideo (Erklärung der Struktur des Projekts und des Code und insbesondere auch als Grundlage für die Bewertung der Softwarequalität dient)
- **[4P je Person]** Selbstbezogener Reflexionsbericht

### Warnung

Nicht realistische Planungen bzw. Aussagen, zum Beispiel in Hinblick auf die Qualitätsmaßnahmen oder die Risiken, führen zu Abzügen bei der Benotung.

Ihre Planung muss die Rahmenbedingungen des Projekts berücksichtigen (z. B. verfügbare Zeit, verfügbare Ressourcen, Vorkenntnisse der Teammitglieder, ...) und realistisch durchführbar sein.

### Wichtig

Jedes Team bzw. Teammitglied muss protokollieren *an welchem Teil des Projekts wer wie viel zum Ergebnis beigetragen hat*. Dies ist notwendig, um am Ende des Projekts eine individuelle Benotung zu ermöglichen, wenn dies erforderlich ist. Hierfür werden die Protokolle herangezogen. Stellen Sie sicher, dass jeder im Team in etwa gleich viel Arbeit an entsprechend bewerteten Teilen des Projekts investiert hat bzw. investieren konnte.

Sollten Sie als Team bewertet werden wollen, so ist dies auch möglich, dann brauche ich jedoch von jedem Teammitglied eine entsprechende individuelle Email (kein CC!), dass die Benotung auf Teamebene gewünscht ist. Ich brauche dann pro Veranstaltung (Projektkonzeption und Projektrealisierung) eine Mail. Die individuelle Protokollierung

der Arbeitszeiten ist jedoch weiterhin erforderlich.

*Der Reflexionsbericht ist jedoch immer individuell zu verfassen und wird immer individuell benotet.*

---

[1] Vorträge: Hinweise und Bewertungskriterien

# Teamaufteilung

3 Gruppen mit je 8 Studierenden mit folgender Aufteilung:

- 2 Entwickler für Apps
- 3 Entwickler für den App Store (Frontend)
- 3 Entwickler für den App Store (Backend)

Diese Aufteilung muss dokumentiert und gelebt werden. Ein Rollenwechsel ist nur in Ausnahmefällen und mit Absprache möglich.

Darüber hinaus sind im Verlauf des Projekts mind. folgende Aufgaben frei zu verteilen:

- Projektmanagement
- Infrastrukturmanagement
- Qualitätsmanagement: Buildautomation, Testmanagement, Testautomatisierung, Codequalität, UX
- Dokumentation: Architektur, Entwicklerdokumentation, Nutzerdokumentation

# 1. Projektkonzeption

# 1. Projektkonzeption - 19.11.2025

## Teil 1 - Einführung

## Teil 2 - Requirements Workshop zur Erhebung der User Stories

**Je Gruppe:** ca. 60 Minuten, um User Stories zu erfassen.

## Teil 3 - User Story Diskussion

**Je Gruppe:** Darstellung und Diskussion der User Stories

### ▲ Achtung!

Es wird erwartet, dass alle Gruppen basierend auf den Erkenntnissen ggf. Ihre User Stories anpassen/erweitern.

## 2. Projektkonzeption - 24.11.2025 [Online]

**BBB:** <https://bbb.dhbw.de/rooms/eic-dx8-r7g-ioa/>

- Kurze Wiederholung von Aspekten des Softwareprojektmanagements

- Aspekte der Projektdurchführung

Gruppenindividuelle Unterstützung bei der Ausarbeitung der initialen Architektur und des Technologiestacks. Beantwortung von Rückfragen.

Je Gruppe:

- Vertraut machen mit dem Technologiestack
- Prototyping / Entwicklung eines Spike
- Ausarbeitung der initialen Präsentation und der Abgabe

### 3. Projektkonzeption - 01.12.2025

#### Teil 1 - Präsentation und Vorführung

(Je Gruppe max. 45 Minuten)

#### **Vorführung**

- Lokale OpenStack Umgebung

#### **Präsentation**

##### **Bewertete Bestandteile:**

- **[3P]** Projektorganisation: Teamname, Teamaufteilung, Zuständigkeiten, Kommunikationswege, Meetingstruktur, Stakeholdermanagement, Hardware-Ressourcen, ...
- **[2P]** gewählte Lizenz(en): für das Endprodukt und die Apps gewählte Lizenz(en) mit Begründung
- **[4P]** Risikomanagement: d. h. identifizierte Risiken und Maßnahmen zur Risikominimierung/-vermeidung

##### **Unbewertet Bestandteile:**

- initiale Architektur inkl. Technologiestack für Apps und App Store
- UI Konzept für den App Store
- Build-Prozess und CI/CD-Pipeline (für App Store und Anwendungen)
- Erfasste (nicht-)funktionale Anforderungen
- Geplante Qualitätsmaßnahmen

#### **Abgaben** (zum 1.12.2025, 7:00Uhr über Moodle)

- Es ist ein Dokument (PDF) abzugeben, dass *die obigen bewerteten Punkte abdeckt*.

Aus dem Dokument muss der Teamname und der Hauptansprechpartner für die Projektleitung/Administratoren klar ersichtlich sein. Diese Person muss garantieren, dass sie - sobald Sie die offizielle OpenStack Umgebung nutzen im zweiten Teil des Projekts - auch für Rückfragen zur Verfügung steht. Die Reaktionszeit auf Mails sollte max. 24 Stunden an Werktagen betragen.

- Die Präsentation (PDF)

Die Anzahl der erreichten Punkte ergibt sich aus der Präsentation und dem Dokument zusammen.

#### Teil 2 - gemeinsame Verfeinerung der User Stories



## 4. Projektkonzeption - 08.12.2025 [Online]

Gruppenindividuelle Unterstützung und Beantwortung von Rückfragen.

**BBB:** <https://bbb.dhbw.de/rooms/eic-dx8-r7g-ioa/> (Optional)

Je Gruppe:

- Verfeinerung der initialen Architektur
- Fortsetzen des Prototypings
- Ausarbeitung der Abschlusspräsentation und der Endabgabe

# 5. Projektkonzeption - 15.12.2025

## Teil 1 - Abschluss der Konzeptionsphase

(Je Gruppe max. 45 Minuten zzgl. Diskussion/Rückfragen.)

### Vorführung

- [1P] Aktueller Prototyp des App Stores

### Präsentation

*(Sie können die Reihenfolge der folgenden Punkte bei der Präsentation anpassen.)*

- [4P] Präsentation der Technologien des (aktuellen) Technologiestacks inkl. Begründung der Notwendigkeit/Sinnhaftigkeit
- [4P] Architektur des App-Stores (insbesondere mit Hilfe von Standardkonformen UML Diagrammen)
- [4P] UI Konzept (z. B. Wireframes oder Mockups)
- [2P] Entwicklungsmodell für die Apps
- [2P] Build-Prozess und CI/CD-Pipeline für App Store
- [2P] Build-Prozess und CI/CD-Pipeline für Apps
- [8P] Qualitätsmaßnahmen: Welche sind wann und warum geplant. Wie sieht die konkrete Ausgestaltung aus.
- [4P] nicht-funktionale Anforderungen
- [8P] funktionale Anforderungen (vollständige User Stories inkl. Akzeptanzkriterien)

### Abgaben (zum 15.12.2025, 7:00Uhr über Moodle)

- Präsentation (PDF)
- Vollständige Dokumentation (PDF)
- Dokumentierte Zeitaufwände (PDF)
- [2P je Person]: Reflexionsbericht mit max. 2 Seiten (PDF): Was lief gut, was lief schlecht, was sollte im weiteren Verlauf beachtet werden, was habe ich persönlich gelernt, wie werde ich zu einem erfolgreichen Projekt beitragen.

## Teil 2 - Übergang zum Projekt

**Je Gruppe:** Planung und Präsentation der nächsten Schritte

## 2. Projektrealisierung - Teil 1 (1. Semester)

# 1. Projektrealisierung - 18.12.2025 (Online)

**BBB:** <https://bbb.dhbw.de/rooms/eic-dx8-r7g-ioa/>

## ■ Jede Gruppe:

- Kurze Vorstellung des Standes (erledigte User Stories, nächste User Stories)
- kurze Vorführung des Entwicklungsstandes
- Ggf. (gruppenindividuelle) Unterstützung bei der Projektrealisierung und Beantwortung von Rückfragen.

## 2. Projektrealisierung - 12.01.2026 (Online: 13:15 Uhr bis 15:15 Uhr und 16:45 Uhr bis 18:45 Uhr)

**BBB:** <https://bbb.dhbw.de/rooms/eic-dx8-r7g-ioa/>

### ■ Jede Gruppe:

- kurze Vorführung des Entwicklungsstandes
- Kurze Vorstellung des Standes (erledigte User Stories, nächste User Stories, Build-Prozess, QS-Stand)
- Ggf. (gruppenindividuelle) Unterstützung bei der Projektrealisierung und Beantwortung von Rückfragen.

### 3. Projektrealisierung - 14.01.2026

- Jede Gruppe:

- Präsentation des aktuellen Projektfortschritts

- (Umgesetzte User Stories; geplanter und tatsächlicher Aufwand; Entwicklung der Story Points)

- ggf. Präsentation von unerwarteten Problemen und deren Lösung

- Präsentation der Ressourcenplanung (zur Verfügung stehende Arbeitszeit pro Person) für nächste Iteration

- Besprechen der nächsten Anforderungen/User Stories.

- Jede Gruppe:

- Aktualisierung und Schätzung des Backlogs

- Mitteilung der Ziele für die nächste Iteration

## 4. Projektrealisierung - 19.01.2026 (Online)

**BBB:** <https://bbb.dhbw.de/rooms/eic-dx8-r7g-ioa/>

### ■ Jede Gruppe:

- kurze Vorführung des Entwicklungsstandes
- Kurze Vorstellung des Standes (erledigte User Stories, nächste User Stories, Build-Prozess, QS-Stand)
- Anforderungen an die Zwischenpräsentation klären.
- Ggf. (gruppenindividuelle) Unterstützung bei der Projektrealisierung und Beantwortung von Rückfragen.

## 5. Projektrealisierung - 28.01.2026

### Abschluss der ersten Projektrealisierungsphase

Zwischenpräsentation je Gruppe ca. 45 Minuten inkl. Diskussion/Rückfragen. 18 von 120 Punkten der Endnote ergeben sich aus der Zwischenpräsentation in Verbindung mit den abzugebenden Dokumenten.

#### Vorführung

- [2P] Aktueller Prototyp des App Stores
- [2P] Prototypen zweier Apps (inkl. Darstellung des (zukünftigen) Deploymentprozesses; d.h. es muss noch kein Deployment über den App Store erfolgen.)

#### Präsentation

- [2P] Aktueller Stand der statischen und dynamischen Architektur des App-Stores: Dies sollte im Wesentlichen über Abbildungen erfolgen und kann ggf. auch die Darstellung eines Datenbankschemas beinhalten.
- [2P] Technologiestack des App-Stores und der Apps.
- [1P] Build-Prozess: Es sollen alle Schritte dargelegt werden; inkl. Konfiguration und Zielen. Es muss klar ersichtlich sein, welche Schritte manuell und welche automatisiert durchgeführt werden. Für manuelle Schritte ist ggf. zu begründen warum diese nicht automatisiert wurden bzw. werden können.
- [1P] Stand der QS: Es ist insbesondere der Prozess darzustellen, wie auf Probleme (zum Beispiel beim Merge, Tests, ...) reagiert wird.
- [8P] Backlog: d.h. erledigte User Stories, noch offene User Stories, Schätzung der Story Points, Entwicklung der Velocity während des Projekts.

#### Abgaben (zum 26.01.2026, 12:00Uhr über Moodle)

- Aktualisierte Projektdokumentation (PDF)  
Dies umfasst ggf. auch die Teile, die im Rahmen der Projektkonzeption abgegeben wurden. Weiterhin sind folgende Punkte abzudecken:
  - QS Maßnahmen (d.h. wer macht was wann und warum?)
  - Architektur
  - Technologiestack
  - (vorläufige) Installations- und Konfigurationsanleitung
- Backlog: mit erledigten und offenen User Stories inkl. Akzeptanzkriterien, Story Points und Entwicklung der Velocity

#### Abgaben (zum 28.01.2026, 7:00Uhr über Moodle)

- Präsentation (PDF)





## 3. Projektrealisierung - Teil 2 (2. Semester)

### ▲ Achtung!

Die folgenden Informationen werden zum Beginn des 2. Semester aktualisiert.  
Sie sind aktuell als Platzhalter zu verstehen.

## 6. Projektrealisierung - DD. MMMM 2026

- Noten Zwischenpräsentation
- Überblick über den weiteren Ablauf
- nächste Schritte / User Stories auswählen

Start der ersten Iteration

## 7. Projektrealisierung - DD. MMMM 2026



Online: <https://bbb.dhbw.de/rooms/eic-dx8-r7g-ioa/>

■ Diskussion von Rückfragen

## 8. Projektrealisierung - DD. MMMM 2026

- Präsentation des aktuellen Projektfortschritts (ca. 8 Minuten pro Gruppe)
  - Stand bzgl. User Stories
  - Stand bzgl. QS
  - Stand bzgl. Build-Prozess
  - Herausforderungen
  - ggf. Live Demo
- Iterationsplanung (Wahl der nächsten User Stories)
- Start der zweiten Iteration

## 9. Projektrealisierung - DD. MMMM 2026



Online: <https://bbb.dhbw.de/mannheim/eic-mn5-hvh-7qd>

■ Diskussion von Rückfragen

## 10. Projektrealisierung - DD. MMMM 2026

- Präsentation des aktuellen Projektfortschritts (ca. 8 Minuten pro Gruppe)
  - Stand bzgl. User Stories
  - Stand bzgl. QS
  - Stand bzgl. Build-Prozess
  - Herausforderungen
  - ggf. Live Demo
- Iterationsplanung (Wahl der nächsten User Stories)
- Start der dritten Iteration

## 11. Projektrealisierung - DD. MMMM 2026



Online: <https://bbb.dhbw.de/mannheim/eic-mn5-hvh-7qd>



## 12. Projektrealisierung - DD. MMMM 2026

- Festlegen der abzuschließenden User Stories
- Besprechung der Endabgaben und der Bewertungskriterien
- Diskussion der Anforderungen an die Abschlusspräsentation

### Notenrelevante Bestandteile

#### Zwischenpräsentation:

$\Sigma$  20 (erledigt)

#### Projekt:

$\Sigma$  **50**

■ Sourcecode **25**

■ Build-Prozess **5**

■ Dokumentation: Architektur **10** (4 Punkte davon ergeben sich aus der Abschlusspräsentation.)

■ Documentation: Installation und Konfiguration **5** (2 Punkte davon ergeben sich aus der Abschlusspräsentation.)

■ Reflexionsbericht **5** (pro Person)

#### QS:

$\Sigma$  **10**

■ QS Maßnahmen - Beschreibung **5**

■ QS Maßnahmen - Durchführung **5**

#### Produktvideo:

$\Sigma$  **20**

■ Produktvideo - Inhalt **15**

■ Produktvideo - (technische) Qualität **5**

Start der vierten Iteration

# Projektabschluss - DD. MMMM 2026

## ■ Abschlusspräsentationen (Pro Gruppe 30 Minuten + Fragen)

ca. 15 Minuten Präsentation der Architektur und der Installation und Konfiguration

ca. 15 Minuten *Live Demo* des Produkts

## ■ Diskussion des Standes dieser Iteration und ggf. Anpassung des Scopes.

Die finale Einreichung aller Dokumente etc. erfolgt über Moodle am Ende des Semesters (d. h. am Sonntag nach der letzten Klausur.)

Videos sind per Downloadlink zur Verfügung zu stellen. Die Auflösung sollte dabei (1600-1920) x (1080-1200) auf keinen Fall überschreiten.

### ▲ Achtung!

Dies markiert das Endes der vierten Iteration!