

Projekt-Systementwicklung / Projekt KMI im SoSe 2024

Prof. Dr. Stefan Linus Zander – Dipl. Volkswirt Roland Trefftz – Lars Kleinschmidt (BSc)

Kick-Off-Meeting – 22.04.2024

Agenda

1. Gemeinsame Kennenlernrunde zu Beginn (alle)
2. Allgemeine Informationen zum Modul (S. Zander)
3. Vorstellung der Projekte und Arbeitspakete (S. Zander)
4. Projektassignments (alle)
5. Gemeinsame Q&A-Session (alle)
6. Nächste Schritte (alle)

Gemeinsames Kennenlernen

Damit wir uns einander besser kennen lernen, stellt sich bitte jede/r kurz vor

Für Studierende, sagen Sie bitte kurz ...

- ...in welchem Studienabschnitt sie sich befinden
- ...wo Ihre persönlichen **Interessen** liegen
- ...wo Ihre **Stärken** liegen
- ...welche **Erfahrungen** Sie mitbringen
- ...mit welchen **Erwartungen** Sie in das Modul gehen

Beteiligte Personen (1/2)

Hochschule Darmstadt

- **Prof. Dr. Stefan Linus Zander** (Modulverantwortung und Lehrveranstaltungsleitung)
 - Email: stefan.zander@h-da.de
 - Website: <https://fbi.h-da.de/personen/stefan-zander>
- **Lars Kleinschmidt** (Technische Projektleitung – CTO)
 - Email: Lars.Kleinschmidt@stud.h-da.de

Klinikon GmbH

- **Dipl. Volkswirt Roland Trefftz** (fachliche Projektleitung und Stakeholder)
 - Email: trefftz@klinikon.com
 - Website: <https://www.klinikon.com/>

Beteiligte Personen (2/2)

Hochschule Darmstadt – Studierende

- Bahammou, Ayoub – ayoub.bahammou@stud.h-da.de
- Bibaj, Leon – Leon.Bibaj@stud.h-da.de
- Boussaguia, Mohamed – Mohamed.Boussaguia@stud.h-da.de
- Conzen, Tobias – Tobias.Conzen@stud.h-da.de
- Gener Lopez, Teresa – teresa.generlopez@stud.h-da.de
- Noumboussi Tayon, Guirauld – Guirauld.Tayon@stud.h-da.de
- Hübner, Maximilian – Maximilian.Huebner@stud.h-da.de
- Kwasny, Marco – Marco.Kwasny@stud.h-da.de
- Lupus, Max – max.lupus@stud.h-da.de
- Nandha, Sanjeev – Sanjeev.Nandha@stud.h-da.de
- Niazi, Mina – Mina.Niazi@stud.h-da.de
- Spengler, Etienne – Etienne.Spengler@stud.h-da.de
- Tchouankam Ngansop, Paul Arnaud – Paul.A.Ngansop@stud.h-da.de
- Wolf, Samuel – Samuel.Wolf@stud.h-da.de

Allgemeine Informationen zum Modul

Lernziele

- Die Studierenden können eine Fragestellung in einem Teilgebiet der Informatik in einem **Projektteam** **selbständig**, **eigenverantwortlich** und **professionell** bearbeiten.
- Sie erlernen eine **strukturierte** und **professionelle Herangehensweise** und können ihre Ergebnisse in geeigneter Form **dokumentieren**, **präsentieren**, und **reflektieren**.
- Sie wenden ihre bis dahin erworbenen Kenntnisse an und erweitern und vertiefen ...
 - ihre **fachlichen Kompetenzen** in mindestens einem Teilgebiet der Informatik,
 - ihre Kompetenzen im Bereich **Software-Engineering** und **Projektmanagement**,
 - Schlüsselkompetenzen wie **Kooperations-** und **Teamfähigkeit**, **Problemlösungskompetenz**, **Kommunikations-** und **Moderationskompetenz**,
 - **Strategien des Wissenserwerbs**



Quelle: Modulhandbuch des Fachbereichs Informatik mit individuellen Ergänzungen und Anpassungen

Durchführung

- Aufteilung der Studierenden in **Projektteams** entsprechend der Arbeitspakete (siehe Projektplan)
- Jedes Projektteam arbeitet selbständig und eigenverantwortlich an den zugehörigen Arbeitspaketen
 - so lassen sich Arbeitspakete parallel bearbeiten¹
- Jedes Projektteam bestimmt **1 Projektverantwortliche/n (aka WP-Owner)**, die/der ...
 - ... den **Projektfortschritt** kontrolliert
 - ... **Arbeitsergebnisse** einfordert
 - ... als **Ansprechpartner** für das Projektteam und die Auftraggeber fungieren
 - ... **Arbeitspakete** und Sprints koordinieren
 - ... für die **Qualitätskontrolle** verantwortlich ist
 - ... mitarbeiten, aber vor allem Aufgaben **delegieren** können
 - ... an den **WP-Leader-Treffen** teilnimmt
- Projektdurchführung analog dem **Scrum-** und **Design-Thinking** Vorgehensmodell

¹ Durchführung analog dem Scrum-Modell (vgl. <https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-DE.pdf>)

Zeitliches Engagement – eine realistische Betrachtung

Projekt Systementwicklung / Projekt Multimedia ist ein **7,5 CP Modul**.

- Das Bachelor-/Masterstudium sieht **30 CP** an Modulen pro Semester vor
- Bei einer wöchentlichen Arbeitszeit von **42 h** ergibt sich ein zeitlicher Aufwand von **10,5 h** (ohne Pausen) über **14 Wochen** hinweg¹
- Dass ist das zeitliche Engagement, das wir von Ihnen einfordern
- Den zeitlichen Aufwand dürfen Sie frei einteilen

In Summe veranschlagen wir pro Person ein 🕒 **Gesamtzeitbudget von 150h - 160h**





¹ Vorlesungszeit (14 Wochen) + Lernwoche (0 Woche) + Prüfungszeit (1 Woche)

Zeitschiene

Wichtige Deadlines

- Laufzeit: max. 14 Wochen
- **Ende WP-Bearbeitungszeit: 07.07. (EOD) (sharp!)**
- **Ende Migration: 14.07. (EOD)**
 - Projekt ist in den Master gemerged
 - Begleitung der Migration durch dediziertes Team
- **Abschluss Betatestphase der App: 21.07. oder 28.07. (EOD)**
 - Durchführung von User-Tests
 - Bugfixing durch gesamte Gruppe
 - Einreichung und Bearbeitung von Features
 - Testen mit den Apple-/Android-Testtools
- **Offizielles Projektende: 28.07. **

Kommunikation

- via  **Discord**: <https://discord.gg/35n8QnWv>
 - für die Kommunikation der Projektteams untereinander
 - für die Kommunikation mit dem CTO zu technischen Fragen
-  **Email** bei wichtigen und/oder offiziellen Dingen
 - achten sie auf einen aussagekräftigen Betreff
 - bitte alle Verantwortlichen in cc mit aufnehmen
- wöchentliche  **Jour fixe** (ca. 60-90 Min für alle Projektteams) mit den Stakeholdern
 - Durchführung mittels BBB (vFBI/002)
 - Pflichttermin für alle Beteiligten
- wöchentliche  **WP-Leader-Treffen** mit dem CTO
 - verpflichtend für alle WP-Leader (ggf. Ersatzmitglied bestimmen)
 - Besprechung von technischen / fachlichen Entscheidungen und/oder Problemen bei der Projektdurchführung

Entwicklungsplattform und Dokumentation

GitLab

- Das we3ve-Projekt hat **6 GitLab-Repos**; die Studierenden werden von uns dort eingeschrieben
- dort werden Programmcode und Dokumentation (Deliverables, Research, Transcripts etc.) gehostet
- Zus. Repos bitte direkt selbst erstellen und die Teammitglieder hinzufügen

Moodle

- Link zum Moodlekurs: <https://lernen.h-da.de/course/view.php?id=17896>
- offizielle Dokumentationen werden über **Moodle** bereit gestellt
- ebenso wie ergänzende Materialien

Taiga

- offizielles **Projektmanagementtool** ab diesem Semester
- Bitte nutzen Sie dieses für die **Projektplanung** und **Projektdokumentation** (APs, Tasks, DoD, Backlog, Issues, Meetings etc.)
- Link: <https://taiga.fgmmgr.users.h-da.cloud/project/we3ve/>

Bewertungskriterien

In die **Gesamtbewertung** fließt ein

- ... das von Ihnen gezeigte **Engagement**
- ... die von Ihnen erbrachten **Ergebnisartefakte**
- ... **Qualität** und **Umfang** der erbrachten Ergebnisartefakte
- ... Ausfüllen besonderer **Rollen** im Projekt
- ... Ihre **Sichtbarkeit**
- ... die Rückmeldungen der **Stakeholder**
- ... die individuelle Gewichtung der **Beteiligungsmatrix**

Die **Beteiligungsmatrix**

- ... ist von jedem Projektteam verpflichtend zu führen
- ... beziffert den **%-tuen Beitrag** eines Teammitglieds
 - d.h., Sie beziffert den Beitragsumfang, den ein Teammitglied zu einem Ergebnisartefakt geleistet hat
- ... ist in Form einer **Matrix** zu dokumentieren
 - bspw. in den jeweiligen Deliverables
- ... wird von den Projektgruppen **eigenverantwortlich** erstellt

Aufgaben der WP-Owner:innen

- Definition der Anforderungen (aka User Stories) in Abstimmung mit den Stakeholdern und dem Team
- Zeit- und Ressourcenplanung (Epics/Sprints/Tasks) in Abhängigkeit des Semesterplans
- Inhaltliche Abstimmung mit dem Team und den Stakeholdern
- Abstimmung von Aufgaben mit dem Team
- Delegation von Aufgaben an das Team
- Ausgestaltung der Epics / User Stories / Sprints / Tasks in Taiga
- Teilnahme an WP-Owner-Treffen
- Rückmeldung an die LVA-Leiter bei Problemen etc.
- Festlegung eines/r Stellvertreter:in (zusammen mit dem Team)
- Kontrolle der Arbeiten und Arbeitsergebnisse

Essentiell ist ein **proaktives, eigenverantwortliches** und **selbstwirksames** Vorgehen!

Aufgaben der Developer:innen

- Definition von User Stories und Tasks
- Zuordnung zu Epics
- Eigenverantwortliche Übernahme und Bearbeitung von Tasks
- Abstimmung mit dem WP-Owner
- Ausgestaltung der Epics / User Stories / Sprints / Tasks in Taiga
- Inhaltliche Abstimmung mit dem Team und den Stakeholdern
- Stellvertretung der WP-Owner
- Dokumentation

Essentiell ist ein **proaktives, eigenverantwortliches** und **selbstwirksames** Vorgehen!

Erweiterung der we3ve-App

Projektdoku: <https://diy.vcd.org/projekte/we3ve>

Übergeordnete Anforderungen an die App


Ziel: Entwicklung einer einfach nutzbaren Car-Sharing-App auf Basis moderner Web-Technologien.

Allgemeine Anforderungen

- Lauffähig im Web und auf mobilen Endgeräten
- Verwaltung von NutzerInnen, Buchungsreservierungen und Fahrzeugen
- Individualisierbares Dashboard mit div. Übersichten (Buchungen / Nutzung / gefahren Kilometer / Auslastung / etc.)
- Einfache und vielfältige Reservierungsmöglichkeiten (bspw. über einen gem. Kalender / via Email etc.)
- Erforschung von Schnittstellen zur Integration von Telemetriedaten und anderen technischen Merkmalen (Ladezustand, Kilometer etc)
- Einfachste Nutzung und Administration
- Hohe Code- und Dokumentationsqualität

Die konkreten Anforderungen müssen in Stakeholder-Interviews mittels User-Research und Requirements-Engineering-Methoden erforscht und erarbeitet werden.

Übersicht der Arbeitspakete – neu

- WP: Bezahl- und Abrechnungsfunktion (RT)
- WP: Erforschung von Möglichkeiten, Telemetriedaten aus Fahrzeugen auszulesen und in der we3ve-App einzusetzen (RT/LK)
- WP: Design und Deployment einer Website (SZ/RT/LK)
- WP: Guidelines für Deployment im Play- und App-Store (SZ/RT/LK)
- WP: Automatisiertes Deployment in App-Stores und auf Server (LK)
- WP: Entwicklung von automatisierten Tests für Backend und Frontend (LK)
- WP: Bugfixing und Integration (LK) 

WP: Bezahlfunktion (RT)

Im WS23/24 wurde ein Research gemacht. Ergebnis: Einbindung von Paypal als Bezahlplattform zwischen Nutzer an Fahrzeug-Halter (keine fiskalische Einbindung von we3ve.de) empfohlen. Weitere interessante Optionen (im nachhinein entdeckt) sind secupay.com oder qwist.com – beide können auch Lastschrift-Transaktionen tätigen und sind vermutlich preiswerter als Paypal.

- **Ziel**

- Implementierung von zwei verschiedenen In-App-Bezahlungsmöglichkeiten durch einen zugelassenen Finanzdienstleister. Die gesamte Transaktion erfolgt dabei auf seiner Bezahlplattform. We3ve stellt nur die Transaktionsdaten (Namen, Bankdaten, Euro-Betrag) zur Verfügung.
- Voraussetzung für die Bezahlung ist die Generierung eines Fahrtenbelegs:
 - Einzeltransaktion je Fahrt (micropayment) oder
 - Sammeltransaktion (z. B. einmal monatlich) mit Auflistung aller Fahrten. Dieser Beleg ist im WP „Abrechnungsfunktion - Fahrtenbeleg“ zu generieren. Die Anforderungen für den Beleg erstellt WP Fahrtenbeleg.
- Opt-Out-Regelung zur Bezahlfunktion

- **Definition of Done**

- Der Bezahlweg ist mit drei verschiedenen Usern im mock up getestet und abgestimmt worden.
- Zwei verschiedene Bezahl-Funktionen (Paypal und noch ein weiteres) sind zur Bezahlung in der App integriert.
 - Mit Integration des WP in die finale Version können die Tester auf drei verschiedene Arten ein micropayment machen:
 - Paypal, weitere Alternative, Bezahlung außerhalb der App (Opt out-Zahlung) – diese ist auch dokumentiert.
 - Auf die jeweilige Rechtslage wird der Benutzer mit Buchung der Bezahlung hingewiesen.
 - Es gibt in der App (gern auch auf der Homepage) eine Anleitung zur Einrichtung der neuen Bezahlfunktionalität
- Test & Dokumentation

WP Abrechnungsfunktion / Fahrten-Beleg (RT)

- **Ziel**

- Fahrten-Beleg: Der Fahrzeug-Halter legt die Art der Bezahlung fest und damit auch, ob je Fahrt/je Abrechnungsperiode ein Fahrtenbeleg bzw. eine Sammelrechnung generiert wird.
- Der Fahrten-Beleg ist Grundlage der Bezahltransaktion (WP Bezahlen). Eine evtl. In-App-Bezahlung ist im Reporting zu kennzeichnen.
- Reporting-/Abrechnungsfunktionalität – Zu den gesammelten Fahrten je Fahrzeug und Halter wird eine Reporting- und Abrechnungsfunktionalität mit verschiedenen Export-Funktionen erarbeitet.

- **Tasks**

- Verschiedene Alternativen für Fahrtenbeleg/Reporting & Controlling incl. Export erarbeiten
- User-Research: welche Variante wollen Fahrer/Fahrzeughalter und welche Daten/Kennzahlen?
- Mock up Testing: Mit zwei-drei Test-Nutzern wurde die UX getestet und vereinbart.
- Umsetzung von Fahrten-Beleg und Reporting (nach Fahrer/Halter/Nachbarschaftsadmin/DSGVO)

- **Definitions of Done**

- User-Research: Belege & Reporting für Driver und Owner
- Umsetzung von Belege, Reportings und Export
- Test & Dokumentation

WP: Telemetrie-Daten aus Fahrzeugen auslesen und Ihr Einsatz in we3ve

Im WS 23/24 wurde ein Research zu diesem Thema gemacht: Android Auto wurde als favorisierte Telemetrie-Option festgelegt. Zu iOS gibt es außer der händischen Eingabe noch keine automatisierte Lösung.

- **Ziel**

- Bereitstellen von Kennzahlen in we3ve:
 - km-Stand (Fahrt Anfang, Fahrt Ende, Differenz: gefahrene km)
 - Akkustand in we3ve-App, Rest-km lt. Akku
 - Falls keine Telemetrie möglich ist, dann manuelle Eingabe der km zu Beginn und Ende der Fahrt (+ gefahrene km)
- Opt-out durch Fahrzeughalter für sein Fahrzeug möglich
- Übergabe dieser Daten an WP Fahrten-Beleg/Abrechnungsfunktionalität

- **Tasks**

- User-Research: mit mehreren Nutzern im Mock up die Nutzung/Führung durch diese Funktionalität gehen.
- Umsetzung beider Alternativen

- **Definition of Done**

- Umsetzung der Ergebnisse (mindestens händische Alternative)
- Ergebnisse können in WP Fahrten-Beleg/Abrechnungsfunktion übernommen werden
- Test & Dokumentation

WP: Design und Deployment einer Website & Guidelines für Play- und App-Store Deployments (SZ/RT/LK)

- **Ausgangssituation**

- Für die App gibt es mittlerweile eine eigene Domain <we3ve.de>. Aktuell ist dies das frontend für die web-app.
- Ziel ist, eine Homepage zur Vorstellung des Projekts und Erläuterung der wichtigsten Funktionen/Voraussetzungen (bspw. Einrichtung einer eigenen Backend-Instanz etc.)
- Schwerpunkt auf guter Maintenance.

- **Tasks**

- Research der verschiedenen Web-basierten Dokumentationsalternativen (Hugo / Mkdocs / Read the Docs / weitere)
- Research: Was benötigen wir für die Einstellung der App in App- und Playstore?
- Implementierung und Deployment der Website
- Erstellung der Inhalte für App- und Playstore und Einstellung der App
- Anfertigung der Benutzungsdokumentation

- **Deliverables / Resultate**

- Die App/das Projekt hat eine eigene. Homepage für Nutzer (Download aus dem App-Store) - für IT-Affine (Aufsetzen eines eigenen Backends).
- Im Google Play-Store als auch im Apple App-Store ist die App und ihre Funktionalität attraktiv für den Nutzende dargestellt und lädt zum Download ein.
- **Benutzungsdokumentation** (inkl. Vorgehen bei Anpassungen der Website etc.) (Deliverable 1)

WP: Usability-Verbesserungen + Bugfixing + Migration (LK/SZ/RT)

- **Ziel**

- Optimierung der Nutzbarkeit der App durch empirische und heuristische Usability-Evaluierungsmethoden
- Eliminierung von Bugs etc.
- Integration der Arbeitsergebnisse der anderen WPs in den `main`-Branch

- **Tasks**

- Sammlung aktueller Ansätze aus der Usabilityforschung und Testen der App gegen diese Ansätze (=heuristische Usability-Evaluierung)
- Erarbeitung von Verbesserungsmöglichkeiten
- Implementierung der Verbesserungen im Abstimmung mit den Dev-Teams
- Erforschung wie die Flutter-App weitestgehend automatisiert getestet werden kann
- Entwicklung einer repräsentativen Menge von automatisierten Testfällen

- **Resultate / Meilensteine**

- Sammlung und Bewertung aktueller Usability-Evaluierungsansätze mit Verbesserungsvorschlägen für die App (Deliverable 1)
- Dokumentiertes Interview mit den Power- und ggf. neuen Nutzenden (u.a. Jörg Kasper, Roland Trefftz) (Deliverable 2)
- Umgesetzte Verbesserungen
- Implementierung der abgestimmten automatisierten Testfälle

Optionale Arbeitspakete

WP: Erforschung von Sicherheitsaspekten für das Deployment & Veröffentlichung der App (RT) (offen; später; Prio 2)

- **Ausgangssituation**

- Im letzten PSE-Carsharing-Projekt (SS 23) wurde das Backend von Firebase (Google) auf einen ExpressJS Server umgestellt.
- Die damaligen Entwickler weisen darauf hin, dass sie sich in der Entwicklung des neuen backends nicht mit möglichen intrusion/attack-Szenarien durch bössartige Dritte beschäftigt hätten. Vorteil von Firebese als backend-as-a-service ist, dass man sich um solche Funktionalitäten nicht zu kümmern braucht, da dies vom Dienstleister (hier: Google) abgedeckt wird.
- Bei einem eigenen backend stellen sich sehr wohl Fragen nach der Absicherung ggü. Eindringen von Dritten, Schadsoftware, ransomware, etc.

- **Tasks**

- Research: Identifiziere die eindeutigen Angriffs-/Infektionszenarien, die ein Dritter auf die frontend/backend-Kombination starten kann.
- Mit welchem Aufwand kann ich welche Verteidigung/Abwehr betreiben, um ein Eingriff/Eindringen zu vermeiden.
- Schätze den Aufwand ab, der mit der entsprechenden Verteidugng/Abwehr einher geht.
- Setze die effektivsten Verteidungs-/Abwehrmechanismen um.

- **Deliverables / Resultate**

- Darstellung wesentlicher Bedrohungsszenarien (Research) - incl. Priorisierung und Maßnahmenkatalog
- Die wesentlichen (und darstellbaren) Maßnahmen sind umgesetzt.

WP: Telemetriedatenerhebung über die App-Nutzung (LK) (3-4 Personen; später; Prio 2)

- **Ausgangssituation**

- Mit Erweiterung der Funktionalitäten sollte es auch interne Auswertungsmöglichkeiten zur Nutzung der unterschiedlichen Funktionalitäten, Fehlermeldungen/Abstürzen, etc. geben.
- Beispielfragen: "Welche Funktionalitäten werden wie genutzt? Wird etwas gar nicht benutzt? Was sind die "Kernfunktionen"?" oder "Wann stürzt die App ab? (inkl. Fehlermeldung)"

- **Tasks**

- Research der wichtigsten Kennzahlen
- Research der technischen Möglichkeiten zur Telemetriedatenerhebung
- Umsetzung der Telemetriedatenerhebung
- Visualisierung der Telemetriedaten

- **Resultate / Meilensteine**

- Dokumentation der ermittelten Grundlagen (Wichtige Kennzahlen etc.) und der technischen Möglichkeiten
- Implementierung

WP: Umsetzung der Pushbenachrichtigungen (insbesondere bei iOS)

(LK) (2-4 Personen; bel.; Prio 2)

- **Voraussetzungen**

- Entscheidung ob iOS-App-Version oder nicht
- Umsetzung der iOS-Version + Developeraccount
- Zum Testen der Push-Benachrichtigungen ist ein Developeraccount notwendig

- **Ausgangssituation**

- In der 1. Version der App (Flutter-App mit Firebase als backend) gab es die Funktion einer integrierte Push-Benachrichtigung zu verschicken. Mit Umstellung auf das neue Backend musste diese Benachrichtigung auf einen schlanken e-mail-Client umgestellt werden. Dies bedeutet einen Bruch in der Kommunikation.

Probleme:

- a. Firebase nicht erwünscht
- b. Wie können fremde Nachbarschaften (mit eigenen Servern!) Push-Nachrichten senden?

- **Tasks**

- Research: welche technischen Optionen gibt es? (Frei oder kostenpflichtige Lösungen / Welche Vor- und Nachteile gibt es)
- Implementation der gewählten Lösung

- **Deliverables / Resultate**

- Dokumentierung der Research-Ergebnisse
- Gewählte Lösung ist implementiert

Fragen ?

Projektmanagement

Projektmanagement – Teil 1

Da wir analog dem **Agilen Manifesto** vorgehen wollen, anbei ein paar Gedanken zum strategischen und operativen Projektmanagement und der Projektdurchführung. Die folgenden Informationen dienen gleichzeitig als Vorbereitung für die wöchentlichen **Jour fixes**.

- Grundsätzlich: die Projektaufgaben (=Tasks der jeweiligen WPs bzw. User Stories) werden in wöchentlichen **Sprints** abgearbeitet¹
- Die **Detailarbeiten** pro Sprint (Research / Development / etc) legt jedes Team individuell fest
- Jedes Projekt führt ein **Sprint-** und ein **Produkt-Backlog**.
 - Das Sprint-Backlog dokumentiert die einzelnen Aufgaben und Ziele eines Sprints für jedes Team.
 - Im Produkt-Backlog werden die Anforderungen in Form von User Stories und zugehöriger Tasks definiert und dokumentiert.
- Die Sprint- und Produkt-Backlogs werden in Taiga geführt ~~den **Git-Repositories** der jeweiligen Projekte geführt.~~
- Vor jedem Sprint steht die **Planung**; hier werden die Tasks der Arbeitspakete (=APs) aus der Projektbeschreibung in Teilaufgaben (=User Stories und Tasks) zerlegt und in das Sprint-Backlog übertragen (siehe vorheriger Punkt).
- Jeder Sprint wird im **Sprint-Backlog** dokumentiert (Planung & Review).

¹ Nach dem SCRUM Guide sollen Sprints eine zeitliche Dauer von 2-3 Wochen haben; da wir jedoch einen vergleichsweise kurzen und fest-definierten Zeitrahmen haben werden wir wöchentliche Sprints abhalten.

Projektmanagement – Teil 2

- Das **Produkt-Backlog** dient als Basis für das Sprint-Review und ist “Richtschnur” für die Planung des nä. Sprints
- Die MO-JFs sind also **Planungs-** und **Review-Meetings** zugleich.
- Zusätzlich kann dort in von den Teams und Stakeholdern definierten Zeitpunkten eine **Retrospektive** durchgeführt werden.
 - Spätestens nach der Erfüllung eines WPs soll eine Retrospektive stattfinden.
- Zusätzlich bzw. ergänzend zu den JFs finden Zwischenpräsentationen statt – idR. bei der Erreichung eines Meilensteins. Die Präsentationstermine werden durch die Stakeholder festgelegt.

Wichtig: Agiles Vorgehen bedeutet, dass sich die Projektdurchführung den aktuellen Anforderungen und vorherrschenden Gegebenheiten anpasst. Sollte sich im Verlauf des Projektes herausstellen, dass eine Nachjustierung sinnvoll ist, dann werden wir diese gemeinsam besprechen und durchführen.

Projektmanagement – Grundlagen (1/2)

- **Rolle von Epics:**

- Epics sind umfangreiche, grobe Beschreibungen von Funktionen oder Anforderungen, die zu groß sind, um in einem einzelnen Sprint umgesetzt zu werden.
- Sie dienen dazu, komplexe Projekte oder Initiativen in überschaubare Teile zu zerlegen, die dann in einzelnen Sprints bearbeitet werden können.
- Epics helfen dabei, den Gesamtüberblick über die Vision oder die großen Ziele des Projekts zu bewahren, während die Details schrittweise entwickelt werden.
- Sie unterstützen die Produkt-Roadmap und helfen bei der Priorisierung von Arbeitselementen im Product Backlog.

- **Rolle von Sprints:**

- Sprints sind feste Zeitspannen, typischerweise zwei bis vier Wochen lang, in denen ein Entwicklungsteam an der Umsetzung von Aufgaben arbeitet, um ein messbares Inkrement des Produkts zu liefern.
- Während eines Sprints arbeitet das Team an den Aufgaben, die im Sprint Backlog festgelegt wurden, und arbeitet darauf hin, die definierten Sprintziele zu erreichen.
- Sprints bieten einen klaren Zeitrahmen und fördern die Iteration und kontinuierliche Verbesserung, indem sie eine regelmäßige Lieferung von Produktinkrementen ermöglichen.
- Die Sprint-Zyklen erleichtern außerdem die Anpassung an sich ändernde Anforderungen oder Prioritäten, da das Team regelmäßig Feedback erhält und seine Pläne entsprechend anpassen kann.

Projektmanagement – Grundlagen (2/2)

- **Rolle von User Stories:**

- User Stories sind kurze, prägnante Beschreibungen von Funktionen oder Anforderungen aus der Perspektive des Benutzers.
- Sie dienen dazu, die Bedürfnisse der Benutzer zu verstehen und zu priorisieren und den Fokus auf den Wert zu legen, den eine bestimmte Funktionalität für den Benutzer hat.
- User Stories helfen dabei, Anforderungen in leicht verständliche Einheiten zu zerlegen, die vom Entwicklungsteam implementiert werden können.
- Sie unterstützen die Kommunikation zwischen den Stakeholdern und dem Entwicklungsteam und helfen dabei, ein gemeinsames Verständnis der Anforderungen zu entwickeln.

- **Rolle von Tasks:**

- Tasks sind konkrete Einheiten von Arbeit, die erforderlich sind, um eine User Story oder ein Arbeitspaket innerhalb eines Sprints abzuschließen.
- Sie sind detaillierte Aufgaben, die vom Entwicklungsteam identifiziert und umgesetzt werden müssen, um die Funktionalität gemäß der User Story zu implementieren.
- Tasks helfen dabei, den Arbeitsaufwand zu strukturieren und zu planen, indem sie die einzelnen Schritte für die Umsetzung einer Funktionalität aufzeigen.
- Sie unterstützen die Zusammenarbeit im Team, indem sie die Aufgaben aufteilen und den Fortschritt verfolgen.

Insgesamt ermöglichen User Stories und Tasks eine effektive Planung, Umsetzung und Verfolgung von Anforderungen in agilen Projekten. User Stories legen den Rahmen für die Funktionalitäten fest und betonen den Nutzen für den Benutzer, während Tasks den detaillierten Arbeitsaufwand für die Umsetzung dieser Funktionalitäten definieren und den Teammitgliedern helfen, ihre Arbeit zu organisieren.


Projektmanagement – Zusammenfassung

- Epic = Funktionalität (WP in Projektbeschreibung)
- Sprint = Workpackage (Bearbeitung der User Stories eines Epics)
- User Story = Anforderung aus Kundensicht
- Task = Aufgabe (Teil einer User Story)
- Product Backlog = enthält alle User Stories
- Sprint Backlog = enthält ausgewählte User Stories aus dem Product Backlog
- Points = definierten Aufwand / Umfang / Komplexität eines Tasks

Nächste Schritte

- Festlegung der WP-Leader
- Anlegen der Taiga-Accounts
- Definition der User Stories (Anforderungen) und Tasks durch WP-Owner in Taiga
- Lokale Installation des we3ve-Projects
- we3ve-Projekt lokal zum Laufen bekommen
- weitere Einstieg in Flutter
- Kontaktaufnahme mit Stakeholdern um WPs (Epics) weiter auszudefinieren

Jour fixes

- Durchführung wöchentlich zu einem festen Termin
- Dokumentation der Gesprächsinhalte durch einen festgelegten Studierenden in Taiga (wöchentliche Rotation)
- Darlegung der wichtigsten Ereignisse/Ergebnisse in der vergangenen Woche (Retrospektive)
- Darlegung der Aktivitäten in der aktuellen Woche (geplante Aktivitäten)
- Klären von offenen Fragen etc.
- Gemeinsames Controlling / gemeinsame Reflektion der WP-Planung
- Teilnahme für alle Studierenden verpflichtend (mit Web-Cam. )

Retrospektive – Infos und Durchführung

Infos zur Durchführung

Ziel: Subjektive Reflektion von Arbeitprozessen und Methodiken um zukünftige Sprints zu verbessern.

Methode: 4 L¹

♥ What I **LOVED**

- Was lief richtig gut?
- Was lief besser als erwartet?

🎓 What I **LEARNED**

- Was habe ich gelernt?
- Was war eine neue Erfahrung?

⚡ What I **LACKED**

- Was habe ich vermisst?
- Was hätte besser laufen können?

? What I **LONGED** for

- Wonach habe ich mich gesehnt?
- Was hat mir gefehlt?
- Was hätte mir geholfen?

¹ <https://nativdigital.com/retro-methoden-4l/>

Aufgabe

Aufgabe: Jede/r bereitet bitte 1 Slide mit den ausgefüllten Aspekten analog nachstehender Vorlage vor und präsentiert diese im nächsten Meeting.

Vorlage:

What I LOVED	What I LEARNED	What I LACKED	What I LONGED
<ul style="list-style-type: none">• ...	<ul style="list-style-type: none">• ...	<ul style="list-style-type: none">• ...	<ul style="list-style-type: none">• ...
<ul style="list-style-type: none">• ...	<ul style="list-style-type: none">• ...	<ul style="list-style-type: none">• ...	<ul style="list-style-type: none">• ...
<ul style="list-style-type: none">• ...	<ul style="list-style-type: none">• ...	<ul style="list-style-type: none">• ...	<ul style="list-style-type: none">• ...
<ul style="list-style-type: none">• ...	<ul style="list-style-type: none">• ...	<ul style="list-style-type: none">• ...	<ul style="list-style-type: none">• ...

Anschließend werten wir die gesammelten Punkte gemeinsam aus (bspw. mit der Methode "[5X Warum](https://nativdigital.com/retro-methoden-5x-warum/)"¹) und entwickeln Maßnahmen für die kommenden Sprints (bspw. mit der Methode "[Starfish](https://nativdigital.com/retro-methoden-starfish/)"²).

¹ <https://nativdigital.com/retro-methoden-5x-warum/>

² <https://nativdigital.com/retro-methoden-starfish/>