

Projekt-Systementwicklung / Projekt KMI im WiSe 2023

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

Kick-Off-Meeting – 24.10.2023

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 jeder 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

- Gil, Marco Antonio – marcoantonio.gil@stud.h-da.de
- Kutzer, Frederik – Frederik.Kutzer@stud.h-da.de
- Le, Thanh Trung – thanhtrung.le@stud.h-da.de
- Walter, Bastian – bastian.walter@stud.h-da.de
- Bauss, Bela – Bela.Bauss@stud.h-da.de
- Heitmann, Kilian – kilian.heitmann@stud.h-da.de
- Jesske, Benjamin – Benjamin.Jesske@stud.h-da.de
- Kajouj, Yassine – Yassine.Kajouj@stud.h-da.de
- Kieferle, Christian – Christian.Kieferle@stud.h-da.de
- Putranto, Bernhard Ricardo – Bernhard.R.Putranto@stud.h-da.de
- Syed, Aniq Hussain – aniqhussain.syed@stud.h-da.de
- Tanhansen, Vinsky Strauss – Vinsky.S.Tanhansen@stud.h-da.de
- Tran, Dang Quang – dangquang.tran@stud.h-da.de
- Wend, Marwin – Marwin.Wend@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 (=Sprints) parallel bearbeiten
- 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
- Projektdurchführung analog dem Scrum- und Design-Thinking Vorgehensmodell

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 **16 Wochen** hinweg
- 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 **160h**

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

Kommunikation

- via **Discord** (wenn gewünscht): [Link folgt](#)
 - für die Kommunikation der Projektteams untereinander
 - für die Kommunikation mit den Stakeholdern zu technischen Fragen
- oder **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

Entwicklungsplattform und Dokumentation

GitLab

- Das we3ve-Projekt hat 3 GitLab-Repos; die Studierenden werden von uns dort eingeschrieben
- dort werden Programmcode und Dokumentation (Deliverables) 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

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

Erweiterung der we3ve-App

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.

WP 1.1: Bezahl- und Abrechnungsfunktion (RT)

- **Ziel**

- Evaluierung von versch. [state-of-the-art Bezahlfunktionen](#) hinsichtlich des Einsatzes in der App
- In-App-Bezahlen (Google, Apple), Guthabekarten, Kreditkarten, Lastschriften, Rechnungsversand mit Überweisung – (mit/ohne QR-Code)

- **Tasks**

- Erforschung der verschiedenen Wege um eine schlanke Bezahlfunktionen zu realisieren
- Darstellung der digitalen Bezahlkette, wer sind die Player, DSGVO, technischer Aufwand, Wegstrecke zur Implementierung/Umsetzung
- Welche Einschränkungen gibt es durch Vorgaben von Google und Apple?

- **Meilensteine**

- Research: Die gängigen Bezahlmöglichkeiten sind recherchiert, die beteiligten Institutionen mit Ihren technischen Lösungen recherchiert und in einer Punktematrix bewertet.
- Der Aufwand zur Umsetzung der favorisierten Lösung(en) wurde valide geschätzt.
- Technisch darstellbare Möglichkeiten (z. B. Lastschrift) werden auch im Projekt umgesetzt
- Favorisierte, anspruchsvollere Lösungen werden im nächsten Semesterprojekt umgesetzt

WP 1.2: Abrechnungsfunktion (RT)

- Ziel:
 - Implementierung von Abrechnungsfunktionalität verbunden mit verschiedenen Export-Funktionen
- Tasks
 - Erforschung der verschiedenen Wege um eine Abrechnungsfunktion inkl. Export zu realisieren
 - Erforschung, welche Abrechnungsdaten exportiert werden sollen
- Deliverables / Resultate
 - TBD

WP 2: Verbesserter Kalender (RT)

- **Ausgangssituation**

- In der Flutter-App kann über eine Monatsübersicht der Tag ausgewählt werden, in separaten Feldern können Uhrzeit und Dauer der Fahrzeugreservierung angesteuert werden.
- Es gibt aktuell keine Möglichkeit über [Links-Rechts-Pfeile](#) oder [Wischgesten](#) im In-App-Kalender zu navigieren. Nutzer melden, dass sie gern à la Google- oder Apple-Calender auch direkt in der Tagesansicht das Zeitfenster der Reservierung auswählen und festlegen würden.
- Unpraktisch ist auch, dass es für ein/mehrere Fahrzeuge keine [Wochenübersicht](#) gibt, um zu sehen, wo freie Termine sind bzw. welches meine Reservierungen (mit vor-/nachgelagerten Reservierungen) sind.
- Aktuell gibt es keine Möglichkeit, sich seine [eigenen Reservierungstermine](#) in den eigenen Desktop-/Handy-Kalender zu [exportieren](#).

- **Anforderungen**

- Komfortablere Kalender-Funktionalitäten, Unterstützung von Wischgesten
- Wochenübersicht für ein/meherer Fahrzeuge der Nachbarschaft - Identifikation freier Slots
- "Kalender-Abo" in private Kalender (Synchronisation mit privatem Kalender --> Reservierung im pers. Outlook-Kalender und Export in App (=Reservierung von Fahrzeugen über den eigenen pers. Kalender, z. B. ics-Termine exportieren o. ä.)

WP 3: iOS-App (LK)

- **Ziel**

- Erforschung, wie weit man die bestehende Flutter-/Android-App anpassen muss, um diese für iOS nutzbar zu machen
 - Wie sehr muss man die Codebasen trennen um eine gute Funktionalität auf allen Plattformen zu erhalten, während gleichzeitig der Code maintainable bleibt - und ohne riesigen Mehraufwand bei der Weiterentwicklung
- ggf. Auskopplung eines eigenen iOS-Projekts (bei eigenem xCode-Projekt benötigen alle Entwickler-Innen einen eigenen Mac...)

- **Tasks**

- Research und Evaluierung, welchen Entwicklungsaufwand eine eigene iOS-App vs. einer Flutter-iOS bedeutet.
- Darstellung ob eigene iOS-App wirklich Praktikabel und höheren Mehrwert ggü. Flutter-iOS-App bietet (Qualitativ u. sinnvolle Kennzahlen - Erfahrungswerte aus anderen Projekten)
- Umsetzung der aktuellen Flutter-App (mit den Neuerungen im Projekt) auf die iOS-Plattform

- **Meilensteine**

- Die aktuelle App (mit allen Erweiterungen) funktioniert auch flüssig und fehlerfrei im "Cupertino-Design" auf iOS

WP 4: Erforschung von Sicherheitsaspekten für das Deployment & Veröffentlichung der App (RT)

- **Ausgangssituation**

- Im letzten PSE-Carsharing-Projekt (SS 23) wurde das Backend von Firebase (Google) auf einen ExpressJS Server umgestellt. (Damit einher ging eine Reduktion des Dockercontainers um den Faktor 100, eine deutliche Performance-Steigerung/Reagibilität und eine breite Installationsfähigkeit auf unterschiedlichen Servern (self-hosting oder gehosteten Servern).)
- Die damaligen Entwickler weisen darauf hin, dass sie sich in der Entwicklung des neuen backends nicht mit möglichen intrusion/attack-Szenarien durch bösartige 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
- Umsetzung der wesentlichen (und darstellbaren) Maßnahmen.

WP 5: Erforschung (und Umsetzung) von Möglichkeiten Telemetriedaten aus Fahrzeugen auszulesen und ihres Einsatzes in der we3ve-App (RT)

- **Ausgangssituation**

- Wir haben bei der Nutzung von eFahrzeugen die Herausforderung, dass der **Ladestand des Akkus** für den nachfolgenden Sharer eine kritische Größe zum Erreichen seines Ziels ist. Gern würden wir die **bordeigenen Daten** jedes Fahrzeugherstellers (insbesondere km-Stand und Akku-Stand/Reichweite bei e-Autos) in die App einspielen, um die Routenplanung verlässlicher zu gestalten. Da jeder Hersteller seine eigene Plattform entwickelt, ist eine herstellerspezifische Programmierung aufwändig. Seinerzeit wurde die Möglichkeit untersucht, einen Dongle mit OBD2-Schnittstelle (Service-Dongle für Werkstätten zum Auslesen der Telemetriedaten) permanent am Fahrzeug zu installieren und darüber in die App auszulesen. Diese wurde jedoch wieder verworfen, da unpraktisch, nicht sehr verlässlich (mechanische Beschädigung), keine Absicherung gegen externes Eindringen (Bluetooth, etc.) gegeben, u.a.m. Gesucht wird eine Möglichkeit, die **Daten** (idealerweise hersteller-/plattformübergreifend) **auszulesen** und sie für die Zwecke der App (verbleibende km, Ladedauer des Akkus, km-Stand zu Start und Ende der Fahrt) zur Verfügung zu stellen. Sollte das nicht realisierbar sein, gibt es die Möglichkeit über **open-source-maps** die gefahrenen km (zurückgelegte Wegstrecke) verlässlich für die Planung und Abrechnung darzustellen? Diese Daten sind Voraussetzung für die Umsetzung von Punkt 1 "Bezahlfunktion"

- **Tasks**

- Bereitstellen der Informationen in der App: km-Stand Fahrzeug, (= km zum Start), km-Stand Fahrtende, Saldo: gefahrene km, Akkustand in der we3ve-App, Rest-km lt. Akku
- Falls keine plattformübergreifende Abbildung der Telemetriedaten i. d. we3ve-App möglich, dann workaround über open-source-map prüfen.
- Umsetzung der geeignetsten Alternative lt. Recherche
- **Erforschung wie solche Daten fälschungssicher erhoben werden können.**

WP 6: Telemetriedatenerhebung über die App-Nutzung (LK)

- **Ausgangssituation**

- Mit Erweiterung der Funktionalitäten sollte es auch interne Auswertungsmöglichkeiten zur Nutzung der unterschiedlichen Funktionalitäten, Fehlermeldungen/Abstürzen, etc. geben.

- **Tasks**

- Welche Funktionalitäten werden wie genutzt? Wird etwas gar nicht benutzt? Was sind die "Kernfunktionen"?
- Wann stürzt die App ab? (inkl. Fehlermeldung)

- **Deliverables / Resultate**

- Research der wichtigsten Kennzahlen
- Prüfung, ob dies in der App (oder per Datensexport) außerhalb erfolgen soll.
- Umsetzung und visualisierung der identifizierten Auswertungsmöglichkeiten

WP 7: Umsetzung der Pushbenachrichtigungen (insbesondere bei iOS) (LK)

- **Voraussetzungen**

- Entscheidung ob iOS-App-Version oder nicht
- Umsetzung der iOS-Version + Developeraccount zum Testen der Push-Benachrichtigungen (=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:
 - Firebase nicht erwünscht
 - Wie können fremde Nachbarschaften Push-Nachrichten senden?

- **Tasks**

- Research: welche technischen Optionen gibt es? Frei oder kostenpflichtige Lösungen
- Implementation der gewählten Lösung

WP 8: Design und Deployment einer Website & Darstellung im Play- und App-Store (SZ/RT)

- **Ausgangssituation**

- Für die App gibt es mittlerweile eine eigene Domain we3ve.de. Aktuell ist dies das frontend für die web-app.
Gedacht ist, eine homepage zur Erläuterung des Projekts. Darstellung der Einrichtung einer eigenen backend-Instanz etc. einzurichten und auch die wesentlichen Funktionalitäten der App zu erläutern.

- **Tasks**

- Erforschung der verschiedenen Alternativen (Hugo / Mkdocs / Read the Docs / etc.)
- Schwerpunkt auf guter Maintenance
- Hauptsprache Deutsch - automatische Übersetzung auf Englisch per ML

- **Meilensteine**

- 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 Nutzer dargestellt und lädt zum download ein.

WP 9: Usability-Verbesserungen + Bugfixing an der App (SZ/LK)

- **Ziel**

- Optimierung der Nutzbarkeit der App durch empirische und heuristische Usability-Evaluierungsmethoden
- Eliminierung von Bugs etc.

- **Tasks**

- Sammlung aktueller Ansätze aus der Usabilityforschung und Evaluierung der App gegen diese Ansätze (=heuristische Usability-Evaluierung)
- Durchführung eines Interviews mit einem Pownutzer (Jörg Kasper) und ggf. weiteren
- Erarbeitung von Verbesserungsmöglichkeiten
- Implementierung der Verbesserungen im Abstimmung mit den Dev-Teams und den Stakeholdern

- **Meilensteine**

- Sammlung und Bewertung aktueller Usability-Evaluierungsansätze mit Verbesserungsvorschlägen für die App
- Dokumentiertes Interview mit Jörg Kasper u.a.

Ursprüngliche Projektplanung und Ergebnisartefakte der ersten Projektphase

Workpackage 1: User Research

- **Ziel:** Benutzungsgruppenspezifische Nutzungsszenarien und -Anforderungen erforschen
- **Task-Liste:**
 - Task 1: Stakeholderanalyse
 - Task 2: Erarbeitung von Interviewleitfäden für die teilstandardisierten Interviews mit den Stakeholdern (=Stereotypen)
 - Task 3: Durchführung und Auswertung der teilstandardisierten Interviews
 - Task 4: Erarbeitung von Anwendungsszenarien und Anwendungsfällen
 - Task 5: Erarbeitung eines Anforderungskatalogs
 - Task 6: Aufteilung der Anforderungen analog dem KANO-Modell und Finden von Begeisterungsfaktoren
- **Milestones:**
 - **Deliverable 1: User Research** mit
 - Dokumentation aller relevanter Artefakte des User Research
 - Dokumentation der Stereotypenanalyse,
 - Stakeholder-Interviews
 - Anwendungsszenarien und Use-Cases,
 - Anforderungsspezifikation mit Begeisterungsfaktoren
 - Anforderungskatalog inkl. Begeisterungsfaktoren

Hinweis: Die notwendigen Informationen zur Durchführung des User-Research finden sich in den Unterlagen des [Moduls "NZSE"](#) von Prof. Dr. Stefan Zander (siehe [Moodle](#))

Workpackage 2: UI-Entwurf & Prototyping

- **Ziel:** Aus den Erkenntnissen des User Research erste UI-Prototypen erarbeiten und mit den Stakeholdern testen um zu einem finalen, akzeptierten Design zu kommen.
- **Task-Liste:**
 - Task 1: Erarbeitung der GUIs
 - Task 2: Konzeption von User-Tests anhand der aus dem User-Research gewonnenen Erkenntnissen
 - Task 3: Durchführung der User-Tests mit den Stereotypen
 - Task 4: Ableitung von Verbesserungen und Überarbeitung des Designs
- **Milestones:**
 - finaler Entwurf der Benutzungsschnittstellen (GUIs)
 - **Deliverable 2: UI-Entwurf (=Designkonzept)**
 - Dokumentation des Designkonzepts für die User-Tests
 - Dokumentation der durchgeführten User-Tests
 - Dokumentation der gewonnenen Erkenntnisse
 - Dokumentation des finalen UI-Entwurfs

Workpackage 3: Technologie-Research

- **Ziel:** Evaluierung der technologischen und architektonischen Umsetzungsmöglichkeiten; System-Grob- und -feinentwurf.
- **Task-Liste:**
 - Task 1: Sichtung aktueller Technologien bzw. Analyse technologischer- und architektonischer Umsetzungsmöglichkeiten auf Basis der Anforderungsspezifikation
 - Task 2: Systemarchitekturentwurf
 - Task 3: Vorbereitung der Implementierungsphase
- **Milestones:**
 - **Deliverable 3: Technologie-Research**
 - Evaluation der technologischen und architektonischen Umsetzungsmöglichkeiten
 - Dokumentation des System-Grob- und -feinentwurfs
 - Auswahl des Technologie-Stacks
 - Festlegung des Systemdesigns

Workpackage 4: Implementierung, Usability-Evaluation, Deployment

- **Ziel:** Implementierung der Anwendung (Frontend & Backend), Testing, Durchführung von Usability-Evaluierungsmethoden und Fertigstellung der Dokumentation.
- **Task-Liste:**
 - Task 1: Implementierung
 - Task 2: Usability Testing mittels heuristischer und empirischer Evaluationsmethoden
 - Task 3: Code- und Doc-Reviews
 - Task 4: Deployment
 - Task 5: Abschluss (Vervollständigung) von Implementierung und Dokumentation
- **Milestones**
 - **Deliverable 4: Usability-Tests**
 - Dokumentation der Usability-Test inkl. Testfällen
 - Dokumentation der Erkenntnisse aus der Usability-Evaluation (=Auswertung)
 - Dokumentation der Verbesserungen
 - Abgeschlossene Implementierung sowie Deployment des Systems
 - Übergabe des Systems sowie der Dokumentation

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) werden in wöchentlichen **Sprints** abgearbeitet
- 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 der APs und der zugehörigen Tasks definiert und dokumentiert.
- Die Sprint- und Produkt-Backlogs werden in 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 zerlegt und in das Sprint-Backlog übertragen (siehe vorheriger Punkt).
- Jeder Sprint wird im **Sprint-Backlog** dokumentiert (Planung & Review).

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.

Jour fixes

Retrospektive – Infos und Durchführung

Infos zur Durchführung

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

Methode: 4 L

♥ 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?

<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") und entwickeln Maßnahmen für die kommenden Sprints (bspw. mit der Methode "Starfish").

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

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