

Programmmentwurf Web Engineering 2

Location-based PWA "loco"

Dokumentation des agilen Projektmanagements

im Studiengang Informatik
mit Schwerpunkt Mobile Informatik

an der DHBW Ravensburg
Campus Friedrichshafen

von

Tobias Welti, Luca Kaiser, Tobias Valentin Jung, Danny Seidel, Andreas Holzapfel,
Sandra Knäfel und Thomas Schmitz



Bearbeitungszeitraum: 26.04.22 - 17.07.22

Dozent: M.Sc. Friedhelm Koch

1 Einleitung

Dieses Dokument dient zur Beschreibung der agilen Arbeitsweise des Projekts "loco". Es wurde nach der SCRUM Methode gearbeitet. Im Folgenden wird die Vorgehensweise genauer erläutert.

1.1 Teamaufstellung

Die aufgeteilte Epic ergab verschiedene Aufgabenbereiche, welche von verschiedenen Teams bearbeitet wurden. Die Teamaufstellung für die einzelnen Bereiche lautet wie folgt:

Produkt Owner:

Friedhelm Koch

UI:

Tobias Welti

Luca Kaiser

API Calls:

Andreas Holzapfel

Map und Routing:

Danny Seidel

Tobias Valentin Jung

Scrum Master und Unterstützung der anderen Teams:

Sandra Knefel

Thomas Enzo Schmitz

Trotz Aufteilung hat jedes Teammitglied überall ausgeholfen, um die Erfahrungen möglichst breit zu fächern und den Teamkollegen unter die Arme zu greifen.

2 Epic

Über die Epic wird die Aufgabe im SCRUM definiert.

Entwickeln Sie eine Web-Applikation, die innerhalb eines Location-Based-Service eine Karte darstellt. Innerhalb der Karte soll eine Position (oder aktueller Standort) mit ihren Geo-Koordination ausgewählt werden können. Über diese Koordinaten soll mittels Reverse-Geocoding der Ort ermittelt und über Wikipedia die entsprechenden Informationen zur Örtlichkeit ausgelesen und visualisiert werden. Anschließend soll die Fahrroute von der gegenwärtigen Position zum ausgewählten Ort dargestellt werden.

2.1 User-Stories

Die Epic wird in kleinere, bewältigbare User-Stories unterteilt.

Beispiele der Unterteilung:

1. Eine Karte darstellen.
2. Eigenen Standort innerhalb der Karte darstellen.
3. Ort in der Karte auswählbar machen.
4. Ermittlung der Koordinaten mittels Reverse-Geocoding.
5. Wikipedia-API-Data-Fetch und Darstellung dieser Daten.
6. Routing bzw. Darstellung der Fahrroute von aktueller Position zum ausgewählten Zielpunkt.

2.2 Aufteilung der User-Stories

Die User-Stories wurden in kleinere Tasks aufgeteilt. Somit waren die Aufgaben leichter bearbeitbar und konnten nach den jeweiligen Stärken der Teammitglieder aufgeteilt werden. Diese Tasks wurden dann auf dem Kanban-Board dargestellt.

2.3 Definition of Done

Bei der "Definition of Done"(DoD) haben wir uns beim unterteilen der Aufgaben auf folgende geeinigt:

1. Code fertig und auf GitHub gepushed
2. Code Review durch Teammitglied oder Pair Programming
3. Coding Guidelines und Styling Standards eingehalten und durch Linter überprüft
4. Der Code wurde mittels JS-Docs verständlich kommentiert
5. Es sind keine kritischen Bugs offen

3 Arbeitsweise

3.1 Sprints

1. 26.04.- 17.05.22
2. 17.05.- 07.06.22
3. 07.06.- 28.06.22
4. 28.06 - 17.07.22

3.2 Sprint 1

Beim allerersten Meeting wurde der Name der PWA ausgewählt, die Definition of Done festgelegt und der Estimation Process mittels Planning Poker durchgeführt. Es wurden weiterhin Risiken für den Prozess sowie die Vollendung der Epic betrachtet und darauffolgend Risikobekämpfungsmaßnahmen wie Timelines, Erinnerungen und Kommunikationsmechanismen eingeführt. Um die Userexperience zu optimieren wurden mehrere Kommilitonen, Eltern und Freunde als Tester für die PWA angeworben und deren Input während dem Arbeitsprozess umgesetzt. Da im ersten Moment noch nicht genug KnowHow für einen richtigen Estimation Process vorhanden war, wurde das Planning Poker mehrmals wiederholt. Exploration Sprints fielen in unserem Fall weg, da die zu verwendenden Technologien bereits bekannt waren.

3.3 Sprint 2

Um einen ersten Eindruck zu bekommen wurde mittels Rapid Prototyping eine erste PWA erstellt und analysiert. Es kam zur Änderung von einigen Ideen des ersten Sprints, was die APIs und Features sowie Tools angeht. Es wurde sich für eine Domain entschieden und die Applikation erstmals auf dieser gehostet.

3.4 Sprint 3

Im dritten Sprint hat der Progress zu wünschen übrig gelassen, da dort die Prüfungsphase stattfand und somit wurde das Meiste in den vierten Sprint übernommen.

3.5 Sprint 4

Im vierten Sprint wurden die meisten Features implementiert. Toggle- und Helpbuttons, also Personalisierungen für den User, die formatierte Ausgabe der Wikipediadaten, sowie die Suche der Orte. Für die UI wurden weiterhin Logos und Icons erstellt und eingebaut. In diesem Sprint wurde auch die Installierbarkeit der Applikation zur Offlinenutzung implementiert, sowie die Performance der gehosteten Version optimiert.

4 Meetings

Das Team traf sich in regelmäßigen Abständen zu Meetings, um den Sprint-Review sowie die weiteren Aufgaben zu besprechen. Anschließend wurde häufig gemeinsam weitergearbeitet.

4.1 Gemeinsame Arbeit

Um außerhalb von den Hackathons im Team zu arbeiten und schnellen Fortschritt zu erlangen, wurden mehrere, lange und kurze gemeinsame Treffen auf Discord verwendet. Hierbei wurden Dinge wie Sprint-Reviews und Weekly's durchgeführt, aber auch bugs gefixed sowie Probleme gelöst.

4.2 Kontakt mit dem Product Owner

Wie in der Scrum-Methodik üblich, hielt der Scrum Master Kontakt zum Product Owner, um Updates zu geben und Fragen zu klären.

4.3 Hackathons

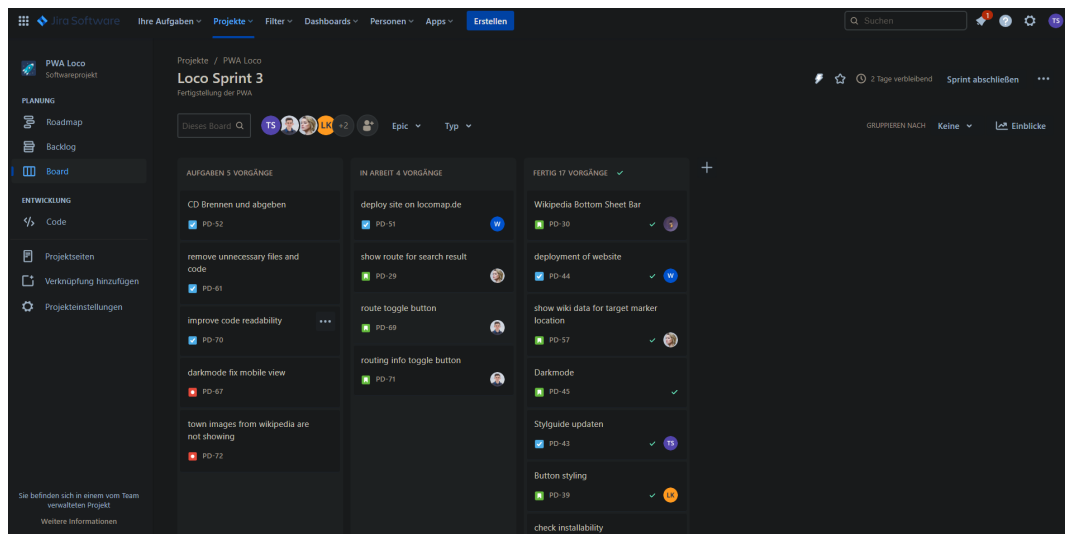
Um sich ungestört mit der PWA beschäftigen zu können, hat sich das Team zu mehreren Hackathons getroffen. Es wurde gemeinsam programmiert, Essen bestellt und Mobilgeräte ausgeschaltet. Bei Hackathons können Probleme direkt mit den Teammitgliedern gelöst werden und die Teambildung gestärkt.

5 Tools

5.1 Jira

5.1.1 Kanban-Board

Um die Scrum Methode grafisch darzustellen wird Jira verwendet, mit einem Kanban-Board verwendet.



5.2 GitHub

Für die Version Control wurde GitHub verwendet. Das Repository ist unter folgendem Link zu erreichen. Hierbei kann die Arbeitsweise mit Branches und Pull-Requests sowie den Lintern eingesehen werden. <https://github.com/DHBW-Hanger/loco>

5.3 Planning Poker

Um die Stories bezüglich ihres Aufwandes und Komplexität zu bewerten wurde Planning Poker verwendet. Hierbei konnten die gemeinsamen Schätzungen zur erhöhten Qualität und Akzeptanz von Schätzungen beitragen, Ideen ausgetauscht werden und verschiedene Sichtweisen betrachtet werden. Desweiteren half die durch das Planning Poker gegebene Gamification Abwechslung in die Planungsphase zu bringen.

5.4 User Journey

Um die User-Experience zu verbessern wurden, basierend auf Abfragen und bereits bekannten Informationen über die Zielgruppe, User Journey Maps erstellt und daraus zu implementierende Features abgeleitet. Beispiele hierfür sind der Darkmode aufgrund von der Anstrengung für die Augen bei hellem Design und der help text um Usern bei Fragen über die Bedienung schnell weiterhelfen zu können.

