



# FLEXED

## Projektdokumentation FlexEd Mobile

Projekt "Mobile Development"  
im SS19/20 an der HAW Hamburg

Juni 2020

**Chau Vu, Clemens Reimer, Julius Schulz**

# Contents

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Vorgehen</b>	<b>2</b>
2.1	Anforderungsanalyse . . . . .	2
2.2	Entwurf und Vorbereitung . . . . .	2
2.3	Arbeitsaufteilung . . . . .	2
2.4	Vorgehensmodell . . . . .	3
2.4.1	Organisation . . . . .	3
2.4.2	Dokumentation . . . . .	3
2.4.3	Umsetzung des agilen Ansatzes . . . . .	3
<b>3</b>	<b>Beitrag der Gruppenmitglieder</b>	<b>4</b>
3.1	Organisation . . . . .	4
3.2	Dokumentation . . . . .	4
3.3	Implementierung . . . . .	5
3.3.1	Erster Meilenstein . . . . .	5
3.3.2	Zweiter Meilenstein . . . . .	5
3.3.3	Dritter Meilenstein bis zum sechsten Meilenstein . . . . .	5
3.3.4	Siebter Meilenstein . . . . .	6
3.3.5	Achter Meilenstein . . . . .	6
<b>4</b>	<b>Fazit</b>	<b>7</b>
4.1	Was gut gelaufen ist . . . . .	7
4.1.1	Flutter, Dart . . . . .	7
4.1.2	Projektumfang . . . . .	7
4.1.3	Kommunikation . . . . .	7
4.1.4	GitLab als Allrounder-Tool . . . . .	8
4.1.5	Teamarbeit . . . . .	8
4.1.6	Arbeitsteilung . . . . .	9

4.2	Was schlecht gelaufen ist . . . . .	9
4.2.1	Zeitplanung . . . . .	9
4.2.2	Aufsetzen der IDE . . . . .	9
4.2.3	Deklarativer Aufbau von Flutter . . . . .	9
4.2.4	Callbacks vs. Changenotifier . . . . .	10
4.3	Fazit . . . . .	10
4.4	Ausblick . . . . .	10

# 1 Einleitung

Als Projekt entstand an einer Schule in Schleswig-Holstein das Konzept von Unterrichtseinheiten mit selbstorganisiertem Lernen (SOL). Dabei haben Schüler die Möglichkeit, sich individuell zu entscheiden, mit welchen Fächern sie sich beschäftigen möchten. Für eine Lehrkraft erschwert dies allerdings den Überblick über den Lernerfolg der Schüler, da der Unterricht viel differenzierter gestaltet ist.

Aufbauend auf einem Projekt aus BAI4-SE2 “FlexEd – Flexible Education”, soll eine mobile Anwendung entstehen, die Lehrkräfte dabei unterstützt, selbstorganisiertes Lernen zu verwalten. Diese soll zu Beginn des Tages die von einem Schüler gewählten Unterrichtsfächer aufnehmen, am Ende des Tages durch eine Bewertung anreichern und anschließend einen Überblick über den Erfolg der Schüler darstellen.

## **2 Vorgehen**

Im folgenden Abschnitt wird das allgemeine Vorgehen im Rahmen des Projektverlaufs in Bezug auf Organisation und Umsetzung zusammengefasst.

### **2.1 Anforderungsanalyse**

Um die Anforderungen des Projektes herauszuarbeiten, haben wir mit einer Lehrkraft der Schule zusammengearbeitet. Im Rahmen von Gesprächen wurden die Anforderungen an die Applikation formuliert und der gewünschte Funktionsumfang definiert. Zusätzlich wurden erste Vorstellungen der grafischen Benutzeroberfläche in Form von skizzierten Mock-Ups erarbeitet.

### **2.2 Entwurf und Vorbereitung**

Aus diesen Anforderungen wurde ein Datenmodell entworfen, welches als Basis der Implementierung diente, und ein Glossar erstellt, um die Begrifflichkeiten für weitere Kommunikation im Team zu klärzustellen. Im Vorfeld wurden die verfügbaren Flutterwidgets betrachtet und eine Vorauswahl getroffen, welche für die Umsetzung des Projektes in Betracht gezogen werden könnten. Auch wurden für bestimmte Teilbereiche des Projektes Libraries betrachtet, die ggf. zum Einsatz kommen sollten.

### **2.3 Arbeitsaufteilung**

Aufgrund der Gruppengröße wurden keine besonderen Aufgaben verteilt, sondern die Applikation wurde in drei Bereiche aufgeteilt: "Erfassen von Zielen", "Bewertung von

Zielen" und "Analyse der Daten". Jeder Bereich wurde dann von einem der Gruppenmitglieder implementiert. Durch dieses Vorgehen war die parallele Bearbeitung der Bereiche überwiegend unkompliziert.

## **2.4 Vorgehensmodell**

### **2.4.1 Organisation**

Durch die Anforderungen des Kurses, wöchentlich einem Stand-Up Meeting beizuwohnen und den derzeitigen Projektstand vorzustellen, bot es sich an, den agilen Softwareentwicklungsansatz zu verfolgen. Unsere Meilensteine passten wir dem wöchentlichen Rhythmus an. Das Team verständigte sich darauf ein Kanban-Board zu nutzen. Die Issues wurden frühstmöglich erstellt und wurden so gut es geht den Meilensteinen zugeordnet. Für die Absprache und Koordination der Ziele der kommenden Woche gab es wöchentliche Voice-Meetings am Sonntag. Soweit es möglich war, stimmten wir unsere Wochenziele auf die Inhalte der Vorträge in der Vorlesung ab.

### **2.4.2 Dokumentation**

Als einziges Werkzeug wurde das Wiki von GitLab eingesetzt. Durch die zahlreichen Plugins kann man nahezu alles unkompliziert in das Wiki einbinden. Ein zentraler Ablageort war damit geschaffen und somit minimierten wir ggf. auftretende Inkonsistenzen zwischen Dokumentation und Implementierung. Anpassungen waren in der Regel komfortabel einzupflegen.

### **2.4.3 Umsetzung des agilen Ansatzes**

Als Team entschlossen wir uns dazu den Fokus auf ein lauffähiges Produkt, welches den minimalen Anforderungen entspricht, zu setzen. Dadurch war es möglich im verfügbaren Zeitrahmen eine Software zu entwickeln, die relativ früh funktionsfähig war. In den letzten Projektwochen hatten wir dann noch genügend Zeit, um das Design komplett zu überarbeiten. Entstanden ist ein lauffähiges Produkt mit einer ansehnlichen und gut bedienbaren Oberfläche.

## **3 Beitrag der Gruppenmitglieder**

Teamarbeit ist ein wichtiger Faktor für den Erfolg eines Projekts. Jedem Mitglied wurden im Rahmen seiner Möglichkeiten geeignete Aufgaben zugewiesen. Die Teamarbeit muss koordiniert werden. Regelmäßige Abstimmungsrunden werden terminiert, Berichte erfasst, Fortschritte dokumentiert und mit anderen Mitgliedern wird kommuniziert. Nachfolgend werden die individuellen Beiträge jedes Teammitglieds zusammengefasst:

### **3.1 Organisation**

Wir haben keine Teamleiter, sondern wir sind alle für ein erfolgreiches Projekt verantwortlich. In den ersten zwei Wochen des Projekts erstellten wir den Projektzeitplan und legten für jede Phase fest, welche Aufgaben erledigt werden. Wir hielten uns gegenseitig regelmäßig auf dem aktuellen Stand. Für die technische Architektur (Datenmodell) sind Clemens Reimer und Julius Schulz verantwortlich. Die ersten Mockups wurden von Chau Vu erstellt.

### **3.2 Dokumentation**

Die Dokumentation wurde im GitLab-Wiki zusammengefasst und jedes Teammitglied schrieb seinen Teil nach der Aufgabenaufteilung. Die allgemeine Beschreibung des Projekts wurde von Julius Schulz formuliert und die Anforderungen von Clemens Reimer.

## **3.3 Implementierung**

### **3.3.1 Erster Meilenstein**

Im ersten Meilenstein haben wir uns mit dem Setup des Flutter Projekts beschäftigt und versucht, Flutter durch verschiedene Tutorials kennenzulernen.

### **3.3.2 Zweiter Meilenstein**

Julius Schulz :

- Implementierung von Utility-Klassen
- Implementierung der Projektstruktur

Clemens Reimer :

- Implementierung von Fach und Daten Typen
- Implementierung Gruppe und Student Klasse

Chau Vu :

- Implementierung der Navigation

### **3.3.3 Dritter Meilenstein bis zum sechsten Meilenstein**

Julius Schulz :

- Implementierung der Seite Bewertung

Clemens Reimer :

- Implementierung der Seite Analyse

Chau Vu :

- Implementierung der Seite SOL Daily Tracking



### **3.3.4 Siebter Meilenstein**

Um sicherzustellen, dass unsere Anwendung weiterhin funktioniert, haben wir Unit-Tests geschrieben. Unit-Tests sind praktisch, um das Verhalten einer einzelnen Funktion, Methode oder Klasse zu verifizieren

Julius Schulz und Clemens Reimer : Implementierung der Unit-Tests

- Tests für SOLCalculator
- Tests für Komponenten
- Unit-Tests für Daten Klassen (Studenten, Gruppe)
- Integration für automatisierte Tests

### **3.3.5 Achter Meilenstein**

- Implementierung des neuen Designs für Seite Bewertung

Clemens Reimer :

- Implementierung des neuen Designs für Seite Analyse

Chau Vu :

- Implementierung des neuen Designs für Seite SOL Daily Tracking
- Verantwortlich für das neue Design der Applikation

## 4 Fazit

Nachfolgend wird der Erfolg des Projektes bewertet, die Lernerfolge betrachtet und ein generelles Fazit bezogen auf das Projekt gezogen.

### 4.1 Was gut gelaufen ist

#### 4.1.1 Flutter, Dart

Alle Gruppenmitglieder sind sich einig, dass Dart eine sehr verständliche und nachvollziehbare Programmiersprache ist. Flutter baut ebenfalls gut auf dieser auf, sodass ein Einstieg sehr leicht fällt.

#### 4.1.2 Projektumfang

Da wir ein konkretes, vorhandenes Problem mit einem echten “Kunden” umgesetzt haben, waren viele Aspekte in der Projektplanung naheliegend (oder “greifbar”). Folglich ließ sich auch der Projektumfang sehr gut einschätzen. Selten wurden Ziele nicht erreicht, und Meilensteine konnten an den Inhalt der Vorlesung angepasst werden.

#### 4.1.3 Kommunikation

Für Absprachen oder Fragen wurde eine WhatsApp<sup>1</sup>-Gruppe für das Projekt erstellt. Außerdem wurden (meist wöchentliche) Meetings über Discord<sup>2</sup> geführt, um Absprachen

---

<sup>1</sup><https://whatsapp.com/>

<sup>2</sup><https://discord.com/>

über weiteres Vorgehen innerhalb der Milestones zu treffen. Die regelmäßige Kommunikation hat geholfen, Probleme und Fragen im Projekt schnell zu klären. Auch die Terminfindung war aufgrund der Größe der Gruppe kein Problem.

#### 4.1.4 GitLab als Allrounder-Tool

Für die Versionierung des Quellcodes, Dokumentation und Projektmanagement wurde als einheitliches Tool die GitLab-Instanz der HAW<sup>3</sup> genutzt. Das Sammeln von allen Informationen über das Projekt an einer zentralen Stelle, hat sich als praktisch und leicht verwaltbar erwiesen.

#### 4.1.5 Teamarbeit

Zwei der drei Gruppenmitglieder haben bereits im vorigen Semester an dem FlexEd-Projekt im Rahmen der Veranstaltung “Software Engineering II” gearbeitet. Bei der Auswahl des Projektes musste deshalb darauf geachtet werden, dass alle Gruppenmitglieder auf dem gleichen Stand bezüglich der Learnings und Funktionalitäten des vorherigen Projektes gebracht werden.

Durch die ausführliche Dokumentation in GitLab und den wöchentlichen Meetings war die Teamarbeit unkompliziert. Fragen und Probleme konnten schnell geklärt werden, und zur Klärung von projektübergreifenden Konzepten wurden Mock-Ups<sup>4</sup>, Tutorials<sup>5</sup> oder Beispielcode<sup>6</sup> erstellt; auch diese Methoden haben dazu beigetragen, Fragen frühzeitig zu klären und die Teamarbeit möglichst angenehm zu gestalten.

---

<sup>3</sup>GitLab Repository des Projektes:

<https://git.haw-hamburg.de/acm167/flexed-mobile>

<sup>4</sup>Mock-Ups der verschiedenen Ansichten der App:

<https://git.haw-hamburg.de/acm167/flexed-mobile/-/wikis/Entwurf/MockUps>

<sup>5</sup>Tutorial zum Erstellen von erweiterbaren Listen mittels Providern:

<https://git.haw-hamburg.de/acm167/flexed-mobile/-/wikis/Architektur/Creating-an-expandable-list-with-the-existing-providers>

<sup>6</sup>Beispielcode für die Nutzung des Repository-Patterns in Views:

[https://git.haw-hamburg.de/acm167/flexed-mobile/-/blob/master/lib/pages/example/repo\\_example.dart](https://git.haw-hamburg.de/acm167/flexed-mobile/-/blob/master/lib/pages/example/repo_example.dart)

### **4.1.6 Arbeitsteilung**

Durch die Aufteilung der App in verschiedene Bereiche, an denen jedes Gruppenmitglied arbeitet, konnten Änderungen an der App auch ohne viele Absprachen erfolgen. Dadurch war jeder in der Lage, Flutter selbst zu “erproben” und die Struktur von Flutter-Apps kennenzulernen.

## **4.2 Was schlecht gelaufen ist**

### **4.2.1 Zeitplanung**

Durch viele andere Praktika von anderen Veranstaltungen, konnte weniger Zeit in das Projekt investiert werden, wie zu Beginn vermutet. Oft gab es Wochen, in denen einfach keine Zeit für die Entwicklung geblieben ist. Trotzdem wurden alle geplanten Ziele erreicht.

### **4.2.2 Aufsetzen der IDE**

Bei mehreren Gruppenmitgliedern gab es Probleme bei dem Aufsetzen von Flutter (Android SDK-Lizenz wurde nicht erkannt). Dieser Fehler wurde durch mehrere Java-Umgebungen erzeugt, und die Fehlersuche war selbst mit Google schwierig.

### **4.2.3 Deklarativer Aufbau von Flutter**

“Easy to learn, hard to master” - Flutter ist zwar leicht zu erlernen, die deklarative Natur ähnlich HTML ist jedoch gewöhnungsbedürftig. Die vielen geschachtelten Elemente lassen schnell den Überblick verlieren, und Refakturierung von Elementen auf niedrigerer Ebene sind zeitaufwendig oder bedürfen einer kompletten Neustrukturierung.

#### 4.2.4 Callbacks vs. Changenotifier

Viele Passagen der App könnten refakturiert werden, indem Callback-Funktionen durch den ChangeNotifier des Provider<sup>7</sup>-Paketes ersetzt würden. Dadurch ließe sich der Code nicht nur aufräumen, sondern würde (wahrscheinlich) auch performanter laufen. Leider bedarf dies (wie zuvor beschrieben) einer kompletten Umstrukturierung der Applikation.

### 4.3 Fazit

Insgesamt sind wir mit dem Verlauf und dem Ergebnis des Projektes sehr zufrieden und die Zusammenarbeit des Teams hat gut funktioniert. Wenn man bedenkt, dass wenig bis keine Erfahrung im Bereich der mobilen Anwendungsentwicklung vorhanden war, verlief die Umsetzung recht problemlos. Alle drei Teammitglieder waren im Rahmen eines in der SE2-Veranstaltung durchgeführten Hackathons schon leicht mit Android App Entwicklung in Kontakt gekommen. Flutter ist deutlich einfacher zugänglich und wir würden in Zukunft auf Flutter zurückgreifen, wenn es um das Thema App Entwicklung geht.

### 4.4 Ausblick

Unter Anbetracht dessen, dass das Projekt für einen realen Kunden entwickelt wurde, wird es auch nach der Veranstaltung weiterentwickelt werden. Durch die fehlende Anbindung an die vorhandene FlexEd-API ist es noch nicht sofort einsatzbereit, könnte aber sogar schon (frühestens) im August in den Live-Betrieb genommen werden.

---

<sup>7</sup><https://pub.dev/packages/provider>