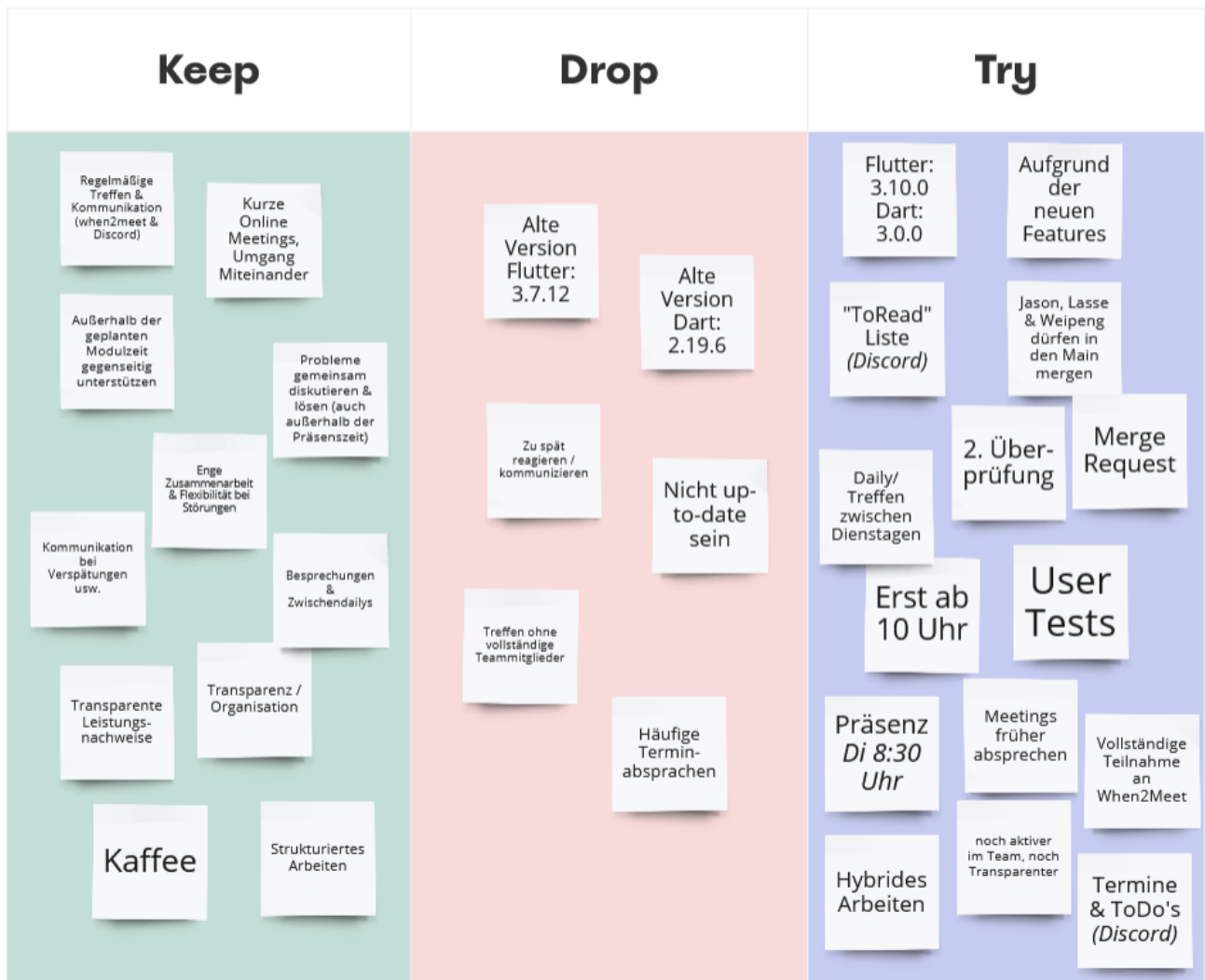


# Planung Sprint I

## Retrospektive



Zur Verfügung stehende Entwicklungszeit

Team-Velocity Sprint 0:

<b>Total</b>	<b>38P</b>
Available Hours	<b>16 h</b>
Velocity	2,375 P/h
Team Velocity	19 P/h

$$2,375 \text{ P} * 8 \text{ Entwickler} = 19 \text{ P / h}$$

# Planung Sprint I

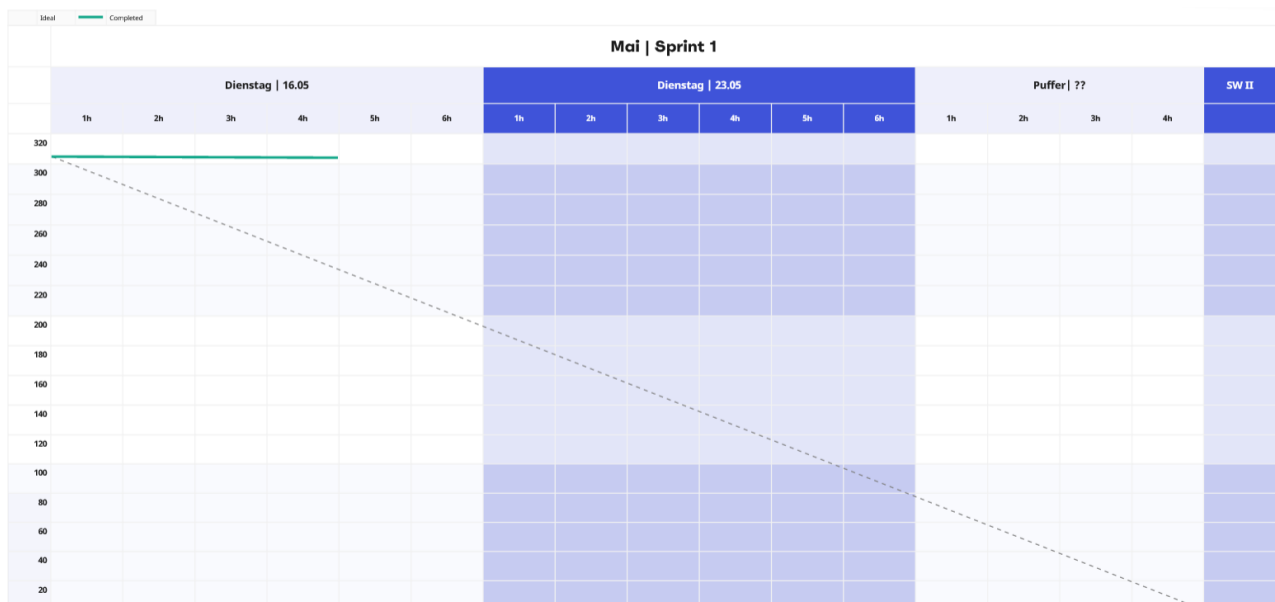
## Maximaler Sprint-Inhalt

Total story points	304 P
Available Hours	16 h
Target	19 P/h

Zur Verfügung stehende Entwicklungszeit \* Team Velocity = max. Sprint-Inhalt

16 h \* 19 P/h = 304 Story Points

## Burndown-Chart



## Definition of Done

- ☐ Überprüfung durch zweiten Entwickler
- ☐ Merge Request gestellt (über GitLab Website - nicht über Terminal)
- ☐ User Test erfolgreich (Funktionalität gewährleistet)
- ☐ Dokumentation der Tests
- ☐ Gemeinsame Code Guideline wurde befolgt
- ☐ Ablage in GitLab → Ticket in GitLab als Feature-Branch entwickelt und hinterher in den Entwicklungsbranch gemerged
- ☐ Ticket ist im Leistungsnachweis vermerkt

# Planung Sprint I

## StyleGuide

### Grundregeln

- Codesprache ist Englisch!
- Eindeutige sprechende Namen!
- Kommentiere so viel wie nötig, so wenig wie möglich!
- try-catch-Blöcke müssen benutzt werden
- Commit-Nachrichten müssen eindeutig sein
- Design Pattern wird eingehalten!

Generell sollen alle **DO** und **DON'T** von den [Dart Style Guide](#) umgesetzt werden.

### Style

Klassen, Enum Typen, Typdefinition (typedef), Typ Parameter <T>, Extensions werden in **UpperCamelCase** geschrieben.

Beispiel:

```
class Klasse { ... }

typedef Predicate<T> = bool Function(T value);
```

Packages, Directories, Quelldateien, Import Variablen Namen werden als **lowercase\_with\_underscores** geschrieben.

Beispiel:

```
my_package
├── lib
│   ├── file_system.dart
│   └── slider_menu.dart
└── ...

import 'dart:math' as math;
import 'package:angular_components/angular_components.dart' as angular_components;
import 'package:js/js.dart' as js;
```

Funktionen, Methoden, Variablen, Attribute, Parameter werden als **lowerCamelCase** geschrieben

Beispiel:

```
var count = 3;

HttpRequest httpRequest;

void align(bool clearItems) {
  // ...
}
```

Konstante werden als **SCREAMING\_CAPS** geschrieben?

Beispiel:

```
const PI = 3.14;
const DEFAULT_TIMEOUT = 1000;
final URL_SCHEME = RegExp('^[a-z]+:');

class Dice {
  static final NUMBER_GENERATOR = Random();
}
```

Abkürzungen gelten als ein Wort.

```
class HttpConnection {}
class DBIOPort {}
class TVVer {}
class MrRogers {}

var httpRequest = ...
var viHandler = ...
var userId = ...
Id id;
```

# Planung Sprint I

## Dokumentation

---

### Kommentare

- Kommentiere auf Sprache?
- Kommentiere in vollen Sätzen
- Benutze keine Block-Kommentare
- Benutze `///` um ein Kommentar für die Dokumentation zu schreiben. Diese können später automatisch in die Dokumentation.
- Kommentiere so viel wie nötig, so wenig wie möglich!

Die Dokumentation muss up to date gehalten werden und am besten auch im Code festgehalten. Das vereinfacht die Wartung.

### Nutzung

---

Falls gewisse Nutzungstechniken auffallen, die man lassen oder nutzen sollte, können diese hier dokumentiert werden.

### Formatierung

Verwendung von Einrückungen für Blöcke und Schleifen  
Verwendung von Klammern für alle Blöcke, auch bei Einzeilern  
Verwendung von Leerzeichen zwischen Operatoren und Variablen  
Verwendung von Leerzeilen zwischen Methoden und Klassen

### Variablen und Typen

Verwendung von `final` oder `const` für unveränderliche Variablen  
Verwendung von dynamischen Typen nur wenn notwendig  
Verwendung von Typ-Annotationen, um den Code klarer und lesbarer zu machen  
Vermeidung von `Var` wenn der Typ offensichtlich ist

### Methoden und Funktionen

Verwendung von sprechenden Namen für Methoden und Funktionen  
Vermeidung von Methoden mit mehr als 15 Zeilen Code  
Verwendung von `void`, wenn die Methode keine Rückgabe hat  
Verwendung von `async/await` statt `then()`-Funktionen

### Flutter-spezifische Empfehlungen

Verwendung von Widgets als eigenständige Klassen  
Verwendung von `MaterialApp` oder `CupertinoApp` als Basis der App  
Verwendung von `StatelessWidget` und `StatefulWidget`  
Verwendung von `ListView.builder()` anstelle von `ListView()`

## Design

---

Falls gewisse Namensgebungen, Formatierungen, Syntax auffallen, die man lassen oder nutzen sollte, können diese hier dokumentiert werden.

## Design Pattern

---

### MVC

---

[MVC Guide] [Model-View-Controller (MVC)]: <https://medium.com/p/1fabe2069b01>

### Benennung

---

Konsistenz in der Namensgebung  
Vermeide reservierte Wörter  
Verwende beschreibende Namen  
Vermeide zu lange Namen

Benennungs-Stile von lokalen-, globalen Variablen, Packages, Library und sonstigem:

Lokale Variablen: aussagekräftige Namen, camelCase (z.B. `userName`)  
Globale Variablen: eindeutige, aussagekräftige Namen, Konstanten in Großbuchstaben (z.B. `MAX_SIZE`)  
Packages: eindeutige, beschreibende Namen, Großbuchstaben und Unterstriche (z.B. `my_first_package`)  
Libraries: beschreibende Namen, Großbuchstaben und Unterstriche (z.B. `my_library`)

### Testen

---

Try Catch muss eingesetzt werden.

# Planung Sprint I

## GIT-Guide

Branch Namen werden in drei Bereiche geteilt und mit Unterstrichen getrennt. Es wird als **UpperCamelCase** geschrieben.

### Struktur:

Feature\_Funktion\_Name

Es gibt einen Master-Branch, einen Develop-Branch und verschiedene Feature-Branches. Von den Features wird nur in den Develop-Branch gemergt, der Develop-Branch ist der einzige der in den Master-Branch mergt. Es kann immer vom Develop-Branch gemergt werden, um seinen eigenen Branch zu aktualisieren.






















Jeder Commit muss mit einem eindeutigen Namen versehen sein.

Beispiel:

Beispiel:

```
| |  
| |  
| | /!  
| |  
| |  
| |  
| | X  
| | * //Feature_LoginSide_Max  
| | /!  
| | \N  
| |  
| | * //Feature_ButtonToSwitchFromStartPageToMainPage_Max  
| | /!  
| |  
| | * //Develop  
| | /!  
| | * //Master
```

## Sprintziel und Backlog

<div>  <b>AEM23T9 Sprint 1</b> </div>	19 issues	ACTIVE	<div> <div>304</div> <div>0</div> <div>0</div> </div>
<div> 16/May/23 2:41 PM • 30/May/23 8:30 PM <a href="#">View linked pages</a> </div>			
<div>             Installierbare FH App, die grundlegende öffentliche Informationen der FH Kiel gebündelt darstellen kann.           </div>			
<div>               AEM23T9-32 Verfügbare App           </div>	<div>             FH Kiel App Grundq...             <div>3</div> </div>		
<div>              AEM23T9-33 Beispielseite, um Stil festzulegen           </div>	<div>             FH Kiel App Grundq...             <div>13</div> </div>		
<div>              AEM23T9-34 Figma Design           </div>	<div>             FH Kiel App Grundq...             <div>20</div> </div>		
<div>              AEM23T9-23 Schnittstelle Quicklinks           </div>	<div>             FH Kiel App Grundq...             <div>8</div> </div>		
<div>              AEM23T9-30 Stundenplanzugriff           </div>	<div>             FH Kiel App Grundq...             <div>0</div> </div>		
<div>              AEM23T9-35 Pdf Ansicht Stundenplan           </div>	<div>             FH Kiel App Grundq...             <div>20</div> </div>		
<div>              AEM23T9-37 PDF Crawler           </div>	<div> <div>40</div> </div>		
<div>              AEM23T9-36 Stundenplan Auswertung           </div>	<div> <div>13</div> </div>		
<div>              AEM23T9-21 Lageplan (Gebäude finden)           </div>	<div>             FH Kiel App Grundq...             <div>0</div> </div>		
<div>              AEM23T9-38 Suche: Wo möchte man hin?           </div>	<div> <div>13</div> </div>		
<div>              AEM23T9-39 Grafische Ausgabe des Ziels           </div>	<div> <div>13</div> </div>		
<div>              AEM23T9-20 Schnittstelle Mensa           </div>	<div>             FH Kiel App Grundq...             <div>3</div> </div>		
<div>              AEM23T9-26 Erinnerungsfunktion           </div>	<div>             FH Kiel App Grundq...             <div>20</div> </div>		
<div>              AEM23T9-29 Gruppen-Zeit-Management           </div>	<div>             FH Kiel App Grundq...             <div>0</div> </div>		
<div>              AEM23T9-40 Gruppen erstellen / Einladen           </div>	<div> <div>30</div> </div>		
<div>              AEM23T9-41 Kalender           </div>	<div> <div>40</div> </div>		
<div>              AEM23T9-31 Einlogfunktion           </div>	<div>             FH Kiel App Grundq...             <div>40</div> </div>		
<div>              AEM23T9-25 Dark Mode &amp; Barrierefreiheit           </div>	<div>             FH Kiel App Grundq...             <div>20</div> </div>		
<div>              AEM23T9-42 Puffer für Unvorhergesehenes           </div>	<div> <div>8</div> </div>		