

### Die seit letztem Meeting durchgeführten Arbeiten umfassen (Sprint 4 abgeschlossen):

- Projektmanagement
  - Aktualisierung der Projektrisiken dieses Sprints
  - Laufend: Aktualisierung der Tickets und des Backlogs, Zeiterfassung pro Ticket
  - Diverse Teambesprechungen zu Arbeitsfortschritt, Aufteilung und Vorgehen
- Projektfortschritt (Programmierung)
  - Bootstrapping-Mechanik: Verbessert, neuer *CommonApplicationBaseModuleLoader* stellt zentrale Dienste wie Storage oder Netzwerk (als *Singelton*) jetzt für alle Features bereit (analog iOS)
  - Module bleiben «im Hintergrund geöffnet», damit z.B. Blog nicht beim Modul-Wechsel neu geladen werden muss
  - In *HsluIApplication/HsluTAApplication* können jetzt Konfigurationen pro Tenant für App-Features gesetzt werden. (Ähnlich zur *HsluApp.swift* bei iOS)
  - *CommonFeatures*
    - Storage-Dienst: Umgesetzt mit Funktionen, Moduldaten in RoomDB zu speichern
    - *TopNav* durch *TabbedDropDownNavigationView* mit den Tabs *News*, *Campus*, *Studies* und *Settings* ersetzt
  - «App»-Features:
    - *Blog*, *Mensa* und *News* mit generischer *WebView* (gemeinsam genutzt) umgesetzt, Funktionalität in eigentlichen Features dadurch nur noch sehr gering
    - *Links*, *Einstellungen* und *Info/About* mit generischer *MultiListView* umgesetzt
- Dokumentation
  - Programmierfortschritt festgehalten
  - Einige Erkenntnisse zum Arbeiten nach AI-First festgehalten, müssen noch im Team besprochen und sauber dokumentiert werden.

### Und dies sind unsere nächsten Arbeitsschritte:

Wir haben in den letzten zwei Wochen viel an den Features programmiert und möchten noch etwas Zeit investieren für Refactoring (Da wir die App vom XML-Projekt übernommen haben, kann wohl noch etwas alter Code bereinigt werden) und uns diesmal etwas mehr auf die Dokumentation kümmern.

Bezüglich der App-Funktionen wären folgende Punkte noch offen:

- Roomsearch: Dieses Feature möchten wir gerne umsetzen, beim nächsten Meeting klären wir am besten wie (HTML oder PDF).
- Events, Parking, Stundenplan, Testate, Anmeldungen, Trails. Diese sind zwar (teilweise) bei iOS umgesetzt, aber wie wir gesehen haben, noch nicht veröffentlicht. Welche dieser Funktionen müssen umgesetzt werden, wo liegt die Priorität?
- Letztes Mal wurde besprochen, für Web-Features auch eine JSON-View zu erstellen. Wir haben dazu auch den neuen Code bei iOS gesehen (und getestet). Die Umsetzung bei Android ist noch nicht gemacht, ist aber schon angedacht («Verzweigung» besteht, View muss noch erstellt werden).
- Auf iOS gibt es Widgets haben wir noch gesehen. (Wird das auch auf Android benötigt?)

**Die drei grössten Risiken zu diesem Zeitpunkt haben wir hier zusammengetragen:**

<b>Risiko</b>	<b>Massnahme(n)</b>
Zeitverzug durch unerwartete technische Schulden (zum Beispiel durch Altcode aus XML-Projekt)	Zeitpuffer für Aufräumarbeiten im Sprint 5 einplanen.
Geplante Features (Roomsearch, JSON-View, etc.) nicht rechtzeitig spezifiziert (z.B. Spezifizierung, ob PDF oder HTML-Ansicht)	Beim nächsten Meeting die Fragen zu den Features klären und festhalten. Feature-Beschreibung dann im Backlog ergänzen.
Inkompatibilitäten zwischen Android-Versionen / Geräten: Es ist schwierig auf verschiedenen Geräten produktiv zu testen (vor allem wenn noch weiterentwickelt wird). Vor allem auf diversen Hardwaregeräten zu testen schwierig, da keine Geräte bereitstehen. Views nur auf Handy-Grössen angepasst (Wie sieht's aus auf Tablets?)	<ul style="list-style-type: none"><li>- In Android Studio gezielt Emulatoren für verschiedene Auflösungen und Android-Versionen anlegen</li><li>- Views vor allem mit flexiblen Layout (z.B. Flexbox) anlegen</li><li>- Nach physischen Testgeräten fragen</li></ul>