**Beta-Version unserer Anwendung (EA3)** Fortschritt seit EA-2

**Allgemein**

Anpassungen an den CSS-Dateien

* Einfügen von CSS-Klassen zur besseren Darstellung von neuen bzw. kommenden Funktionen z.B. Highlighten der Termine im Kalender

Dynamische Effekte in Form von Kreisen mittels Canvas für den Hintergrund eingebaut

* Zusätzlich angeeignetes Wissen ausprobiert, wertet das Layout optisch auf

**Startseite**

Dynamische Darstellung des Fortschritts

* Verknüpfen der Progressbar mit den darüberliegenden Punkten
* Dynamische Darstellung ersetzt reine CSS-Implementierung
* Jeder Punkt stellt nun ein Projekt dar und zeigt einen individuellen Fortschritt an

Aktuell enthält der Code an dieser Stelle noch sehr viel Redundanz. Eine Überarbeitung ist notwendig. Passende Daten sollen zukünftig aus der IndexedDB bezogen werden.

Wechseln zwischen den Tabs

* Ursprüngliche Idee der reinen CSS-Implementierung nun mit JavaScript umgesetzt
* Hierzu wurde sich angeschaut, wie man Elemente mit dem gleichen Klassennamen unterscheiden kann (z.B. bei onclick) und die Eigenschaften *className* und *classList* genutzt

🡪 Dieses Wissen soll auch für den redundanten Code für bei der Fortschrittsanzeige genutzt werden

Gruppen – Anpassung CSS

* Die angeklickten Summaries färben sich beim Anklicken bzw. im geöffneten Zustand der Details nun in unserem Farbschema ähnlich wie bei den Tabs

**Projektseite**

Umbau des zugrunde liegenden Datenmodells zur besseren Persistenz der Daten

* Vorbereitung für die Speicherung in der IndexedDB
* Auch hier nun der Einsatz von Modulen (siehe Kalender), wobei der alte Code noch nach und nach refactored wird

Bei der Projektseite bietet sich der objektorientierte Ansatz eher an, da wir dort mit sichtbaren Objekten im Frontend arbeiten, die sich anhand dessen gut modellieren lassen.  
Mit einem größeren Umbau im technischen Bereich, also nicht direkt im Frontend sichtbar, erhoffen wir uns ein leichteres Vorgehen bei der Speicherung, bzw. Ajax-Operationen.   
Somit auch die Dynamisierung via Ajax und IndexedDb.  
  
Interaktionsmodell der Projektseite in Frage gestellt und Fehler, bzw. fehlende Funktionen in den ursprünglichen Wireframes

* Alternativversuche bei der Funktionsimplementierung ausprobiert, um sich vom Layout her an die Wireframes zu halten
* Bisher keine Forms bedacht, die nun schnellstmöglich nachgeliefert werden müssen
* IDs bei Elementen ergänzt, die dynamisch vom User angepasst werden können

Neue Implementierung des Pie-Charts mittels JavaScript

* Zuvor nur reine CSS-Implementierung
* Aktuell mit hardcoded Werten gefüllt, soll in Zukunft Werte aus der IndexedDB abfragen

**Kalender**

Umbau von objektorientiertem Vorgehen zu funktionalem Ansatz

* Vorbereitung zum besseren Persistieren in der IndexedDB
* JavaScript Objekte direkt speichern ohne viel Umbau
* Einzelne Entitäten haben wir in Module umgewandelt, um eine bessere Erweiterbarkeit der Daten zu erreichen

Da der Kalender sehr datenbezogen ist, haben wir uns hier für einen funktionalen Ansatz entschieden. Zudem hatten wir Lust darauf, andere Programmier-Paradigmen auszuprobieren und somit eine andere Denkweise und Sicht der Dinge zu erlangen.  
Die Persistenz der Termine ist aktuell noch mit Hilfe des *localStorage* realisiert.

**Impressum**

Spielerweiterung

* Erweiterung um ein Zeitlimit, wird bisher nur Neuladen der Seite erneut gestartet
* Infofeld bei Spielende
* CSS-Anpassungen vorgenommen

Fokus für den nächsten Sprint liegt auf der Implementierung der IndexedDB, um somit sinnvoll weitere Funktionalitäten unserer Anwendung umsetzen zu können. Das Feedback aus der Präsenzveranstaltung und uns noch fehlende Anforderungen haben wir dabei im Blick. Zudem möchten wir wieder darauf achten, die Dokumentation anhand von User Stories nicht schleifen zu lassen und nicht nur mündliche Absprachen halten.