

**01.03.2021**

Die Mitglieder Marco, Lawrence, Korab, Blendi und Jethro haben diskutiert was Sie mit den Kunden im ersten Meeting besprechen werden und was der Inhalt des Mail ist.

### Struktur von Meeting

Frage/Aktivität	Zeit
Begrüßung der Mitglieder	~5 min
Projekt vorstellen (Idee, Anwendungen) anhand eines Beispiel	~15 min
Die erste Iteration <ul style="list-style-type: none"><li>- Welche Users</li><li>- User stories</li><li>- Einschränkungen z.B. Browser oder Code Architektur/Qualität</li><li>- Wird Code weiterentwickelt?</li><li>- Wie soll der Code getestet werden; was für ein Coverage?</li></ul>	~20 min
Organisatorisches <ul style="list-style-type: none"><li>- Github collaborator</li><li>- Kunden treffen</li><li>- Vorstellung/Idee von Planning Game</li></ul>	~10 min
Generelle Fragen zum Projekt	~8 min
Verabschiedung	~2 min

### Struktur von Mail

Korab sendet eine Kopie von der obigen Tafel als ein Vorschlag wie das Meeting ablaufen soll und bittet den Kunden ein Beispiel vorzubereiten (Wer der Kunde ist, was für ein Ziel kann er mit dem Produkt erreichen).

Es wurde auch abgemacht, wer wann die Präsentationen vorbereitet. Die Termine findet man auf dem Slack Channel.

Die Rollen der Teammitglieder wurden schärfer definiert.

- Jethro ist für Code Dokumentation, Code Qualität und den Test report zuständig.
- Blendi ist zuständig für den wöchentlichen Statusbericht und für die Risiko analyse. "Advises Team based on previous reports".

**02.03.2021**

## Abilium meeting

Die Mitglieder Marco, Lawrence, Korab, Blendi und Jethro haben die Kunden Jakob Schärer und Severin Zumbrunn getroffen.

### **Geteilte Notizen:**

- Editor View
  - Beginnt mit Root Node des Produkts
    - Firmenlogo
    - Titel des Produkts
    - Produktlogo
    - Beschreibung
  - Erstellen von einem Baum (Knoten hinzufügen, ...)
  - Knoten haben Eigenschaften
    - Titel
    - Text
    - Farbcodierung
      - Hintergrund Zustand der Nachhaltigkeit (Gut, Unbekannt, Schlecht, ...)
      - Umrandung Art der Überprüfung (Besucht, Kontakt, Noch kein Kontakt, ...)
    - Flaggen des Herkunftslandes
    - Kategorie (Rohstoff, Verarbeitetes Produkt, Montage, ...)
  - Bei jedem Knoten können Children hinzugefügt werden
  - Bei jeder Verbindung kann ein Knoten dazwischen geschaltet werden
  - Webbasierend
  - Angular oder ähnlich (Abklären welches Framework sich am besten zum editieren eines Baumes eignet)
  - Export / Import eines XML/JSON zum lokalen speichern.
- View
  - Bild generieren
  - Link zum Bild
  - Widget (tiefer prio)

### **Links aus dem Abilium Meeting:**

<https://visjs.org/index.html>

<https://github.com/visjs/ngx-vis>

[https://threejs.org/examples/#webgl\\_animation\\_keyframes](https://threejs.org/examples/#webgl_animation_keyframes)

<https://visjs.github.io/vis-network/examples/network/exampleApplications/worldCupPerformance.html>

### **Komentar:**

BASE64

Nach einer schnellen Begrüßung, haben die Kunden ihre Firma und das Projekt präsentiert. Sie haben uns gesagt, dass Elektronische Geräte bestehen aus vielen Teilchen, welche aus vielen Bauteilen gebaut sind. Sie haben verschiedene Herkunft, das heißt, dass es schwierig ist Kontrolle über diese Produkte zu haben. Es gibt zum Beispiel ein Fairphone, welche kein Konflikt mit den Materialien hat. Unsere Projekt muss ein Produkt bauen, welche von allen Firmen benutzt werden kann (mit oder ohne Wasserzeichen).

Wie schon gesagt, das Produkt ist ein "plug-in" (oder ein Client-sided Webpage, AKA "Fat Client"), welche graphische Bäume erstellt. Man sollte anfangen mit einem "Root".

Es sollte möglich sein importieren/exportieren ein XML oder JMS file (ein Dateitype reicht es).

Die Kunden haben uns empfohlen, mit guten Grundlagen anzufangen: solide Framework (wie zum Beispiel visjs <https://visjs.org/index.html>) und gute Definition von Objekten. Bei der ersten Iteration sollten wir ein Diagramm haben.

Es sollte einfach für eine Firma, Daten importieren. Ein gutes Ziel für das Ende des Projekts wäre eine gute Weise den Baum gut zu navigieren/visualisieren.

Wir werden mindestens ein Meeting mit dem Kunden pro Iteration haben.

Für unsere Projekte sind die folgenden Testprotokolle wichtig:

- Importieren und Exportieren sind wichtig, deswegen sollten sie gut getestet werden;

- Testen die Funktionen/Logik, die wir implementieren.

Noch wichtig ist, dass es keine Cycle in dem Baum geben sollte.

## **Nach Abilium meeting**

Die Mitglieder Marco, Lawrence, Korab, Blendi und Jethro haben die Anforderungen von Abilium diskutiert.

Bis Freitag sollten wir schon wissen, wie wir was implementieren wollen und über plug-in lesen. Wir haben geplant, am Freitag mit dem Programmieren anzufangen.

Wir haben geplant, unseren Assistenten am Freitag zu fragen, was wir uns für den ersten Iteration planen realistisch ist von technischen und Zeit her.

Am Freitag soll der Key Account Manager einen Mail an unsere Kunde schreiben mit was wir planen für den ersten Iteration.

Wir haben entschieden, dass wir viel mehr Leute im Frontend brauchen und dass alle Angular auffrischen soll. Der ganze Logik soll in den Typescript Dateien sein und fast nichts soll auf der Serverseite sein.

Wir haben uns für mehrere Visualisierungen entschieden und daher haben wir auch entschieden, den Datenstruktur eine möglichst generelle Interface zu geben, sodass es uns ermöglicht, mehrere Visualisierungen daran zu knüpfen. Eine gute Idee wäre "Bäume aus Bäumen" zu machen (man sieht zuerst die wichtige Teile und der User kann die Drucken und er bekommt ein andere Baum).

Ideal wenn es als plug-in wäre.

Bei Risiko Analyse sollte Blendi die Risiken aus dem Scrum meeting nehmen weil der Scrum Master fragt die andere Mitarbeiter wie es läuft und was für Blockaden sie haben. Blendi kann auch aus den Scrum Meetings Information für den Statusberichten bekommen.

Beispiel: <https://app.diagrams.net/>

**05.03.2021**

**09.03.2021**

**12.03.2021**