

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">- Welche Users- User stories- Einschränkungen z.B. Browser oder Code Architektur/Qualität- Wird Code weiterentwickelt?- Wie soll der Code getestet werden; was für ein Coverage?	~20 min
Organisatorisches <ul style="list-style-type: none">- Github collaborator- Kunden treffen- Vorstellung/Idee von Planning Game	~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
 - Webbasiert
 - 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://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. Die 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 wichtigen Teile und der User kann die Drucken und er bekommt ein anderer 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 anderen 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

Team Meeting

Die Mitglieder Marco, Lawrence, Korab, Blendi und Jethro haben über die erste Iteration/Planning Game besprochen (Assistent Adrian war auch dabei).

Feedback von Adrian:

- Abilung meeting Probleme:
 - Team muss den Meeting durchführen
 - Probleme besser Lösen: Falls jemand eine wichtige Rolle hat, jemand soll bereit sein ihn zu substituieren
- Github repository Access an Prof. Studer geben
- Erst eine Basis bauen (z.B. Angular/visjs skills)
- Organisatorisch noch nicht um Performance denken, einfach denken, erste schritte durchführen, um den Projekt zu anfangen → rest kommt alleine

Meeting besprochene Themen:

- Noch nicht um den Workflow/echte Iteration nachdenken
- Ähnliche Basis aufbauen
 - Neue Framework durchsuchen und Einschränkungen beachten (z.B. Lizenz)
 - Kleine Basis auf die Framework bauen
- Wichtig Marco und Jethro: Angular lernen (waren in Backend Team)

Organisatorisch

- Blendi + Marco Präsentation 11 März
- Blendi + Lawrence Protokoll und Risk Analysis Verteilung
 - Lawrence nächste Woche Protokoll, Blendi Risk Analysis, übernächste Woche vertauscht, usw.
- Korab Mail zur Kunde schreiben am Dienstag (9 März) verschieben
- Marco + Blendi Meeting am Sonntag (7 März) für Präsentation
- Alle Teammitgliedern: nach Frameworks durchsuchen für 9 März
- Marco neue Trello board erstellen für Tasks Verteilung
- Blendi für Statusreport verantwortlich
- Repository organisieren für Weekly deliverables und Arbeitsbereich
- Abgabe wöchentliche Deliverables 11 März

09.03.2021

Team Meeting

Frameworks durchsuchen → Wichtig: Lizenz erfüllen

- Wir nehmen vis.js.
- Andere frameworks einsetzen wenn es nötig ist

Antwort Kunden

- Mit Frameworks experimentieren ist fertig
- Ungefähr was in der 1.Iteration kommt (in Trello)
- Korab schreibt einen Mail mit diesen Entscheidungen

Organisation details

- Ideen von Tasks haben wir in "what is needed" in Trello geschrieben
- Die schon gemachte Tasks auf "done" verschoben
- Tasks, die wir für diese Iteration machen wollen, haben wir auf "in progress" verschoben
- Tasks nach Priorität geordnet
- Wie webseite aussieht müssen wir bestimmen
- Wir bearbeiten example.component
- Example component soll kein God Class werden
- Blendi, Marco und Jethro wollen zusammen Pair Programming machen
- Korab und Lawrence wollen zusammen Pair Programming machen
- Link zur Github repository mit den Deliverables wurden an Adrian und Thomas Studer geschickt
- Dependency tree soll erstellt werden

Tasks verteilung?

- Wir haben im Meeting keine Tasks verteilt um die Chance zu verringern, dass jemand einen Task bekommt, wo er nicht so gut weiss, was er tun soll

Deliverables diskutieren -> Abgabe weekly deliverables 11 März

- 3 Risiken wurden identifiziert
- Status von alle ist Mittel

Präsentation besprechen

- Blendi und Marco haben die andere Mitglieder präsentiert
- Die andere haben es kommentiert

12.03.2021

Scrum Meeting + Team Meeting

Zeitplan

Frage/Aktivität	Zeit
Assistent Diskussion (Fragen, Bemerkungen, usw)	~10 min
Scrum Meeting (what I have done, problems...)	~10 min
Next Sprint Planning	~10 min
Präsentation diskutieren	~5 min
Fragen, Bemerkungen, usw	~5 min

Assistent Diskussion

Alle Tasks im Trello aufschreiben
Zuständige für jeden Task finden
Programmiertasks sind nicht die einzige Tasks
Beim Präsentieren sich verkaufen
Einfache Sprache

Scrum Meeting

Erstes Mal Code anschauen war für alle schwierig
Mehr über visjs lernen

Next Sprint Planning

Sprints sind eine Woche lang
Tasks wurden auf Trello verteilt

Präsentation diskutieren

Vielleicht zukünftige Präsentationen Adrian zeigen

Fragen, Bemerkungen, usw

Man kann mehrere Nodes auswählen
Lawrence macht ab jetzt immer den Protokoll
Blendi macht ab jetzt immer den Statusreport und Risikoanalyse

16.03.2021

Scrum Meeting + Sprint planning 2

Zeitplan

Frage/Aktivität	Zeit
Scrum meeting	~10 min
Sprint planning 2: first part (what done? Left?)	~10 min
Sprint planning 2: second part (new tasks/old tasks)	~10 min
Risks for this sprint/week	~5 min
Präsentation Lawrence: Problems?	~5 min
Deliverables for this week	~5 min
Fragen? Bemerkungen?	~5 min

Scrum meeting

Gut: Wir haben viel gelernt über das Framework

Die meisten Basisfunktionen sind schon implementiert

Probleme: Keine Tests, Layout noch nicht bestimmt, Branches geben Mergeprobleme, Implementation von Beispielen aus Visjs ist schwierig

Sprint planning 2: first part (what done? Left?)

Haben klein implementation gemacht

In docs gelesen

Sprint planning 2: second part (new tasks/old tasks)

Blendi: Farben anpassen

Marco und Jethro können pop ups machen

Alle: Prototyp machen bis freitag

Ab Freitag neue Branches nur für neue Komponente

Risks for this sprint/week

Example wird vielleicht zu God class

Dass wir die Basics nicht haben, sowie nodes bearbeiten

Dass wir keine Prototypen haben für der Layout

Präsentation Lawrence: Problems?

Adrian fragen wegen "Analyse 1. Iteration"

Lawrence und Korab machen Pair Programming

Blendi, Marco und Jethro machen zu Dritt Pair Programming, manchmal zu zweit

Deliverables for this week

Die Daten können geändert werden bis Donnerstag

19.03.2021

Scrum Meeting

Zeitplan

Frage/Aktivität	Zeit
Assistant Discussion + Präsentation	~15 min
Scrum meeting	~10 min
Prototype/Layout Discussion	~10 min
Branch Git-hub Struktur diskutieren	~10 min
Git-Hub Deliverables Struktur diskutieren	~10 min
Fragen? Bemerkungen?	~5 min

Assistant Discussion + Präsentation

Analyse ist in handbuch, 2,3 ausspicken

Gruppenorganisation: welche personen pair programming, wie arbeiten

Risikoanalyse anders als deliverables

Struktur von Deliverables

Standardstruktur : master (sollte immer laufen) <- dev <- jedes feature in weitere branches

Donnerstag ein definite branch für Adrian wählen

Demo bewerten: demos zusammenmachen -> Projektanteil

Layout: user sind eher faul

Lieferanten/werkstoffe extern brauchen statt nodes

Nicht zu general: schauen, dass der Kunde einverstanden ist

Scrum meeting

Probieren, Elemente zu editieren

Framework studieren

Prototype/Layout Discussion

Benötigte Options sind geeinigt worden

Korabs Prototyp wird gewählt

Hover über die icon soll Infos geben

Branch Git-hub Struktur diskutieren

Wir nehmen den Vorschlag von Adrian

Frühe Zeitpunkt von network editor als master branch gewählt

Neue Branches machen

ChangeName wird ein component

Unabhängige components machen

Git-Hub Deliverables Struktur diskutieren

Korab schreibt ein Email an Thomas zum unseren Situation klarifizieren

23.03.2021

Scrum Meeting + Iteration 1 Analysis

Zeitplan

Frage/Aktivität	Zeit
Scrum Meeting	~10 min
Sprint planning/retrospective	~10 min
Risk Analysis	~5 min
Iteration 1 Analysis	~10 min
Deliverables for this week	~5 min
Git-Hub Struktur diskutieren (welche branch halten?)	~5 min
Foto vom Team machen	~2 min
Präsentation: Hints, Probleme, Bemerkungen?	~10 min
Fragen? Bemerkungen?	~5 min

Scrum Meeting

Wir müssen der neue Komponentenstruktur verstehen

Risk Analysis

Meeting kann dauern zu lange

Vielleicht sind wir zu langsam

Documentation is schlecht

Die Application ist schwierig zu testen (wegen der Bentzt des Frameworks)

Layout zu ändern wird schwierig

Nicht gute Architektur (keine UML gemacht)

Deliverables for this week

Wir haben uns auf eine bestimmte Zeitplan geeignet

Müssen am Adrian schreiben

Git-Hub Struktur diskutieren (welche branch halten?)

Wir haben uns auf einem dev branch entschieden

Viele Branches wurden gelöscht

Präsentation: Hints, Probleme, Bemerkungen?

red border to team members

Analyse erste Iteration viel zu viel Text

Marco und Jethro haben Lieferanten hinzugefügt

Keine klare Struktur in der Präsentation

Fragen? Bemerkungen?

Wir müssen den Kunden schreiben

Adrian schreiben zum informieren, dass der Deadline für Deliverables verschoben worden ist

26.03.2021

Scrum Meeting + Sprint Planning 3

Zeitplan

Frage/Aktivität	Zeit
Feedback Präsentation + Deliverables	~20 min
Scrum Meeting	~10 min
Sprint planning/retrospective	~10 min
Deliverables Next Week	~5 min
Demo Organisation	~10 min
Osterferien Organization	~5 min
Frage? Bemerkungen?	~5 min

Feedback Deliverables

Deliverables: Foto nicht der gleiche Name haben

Code review: Example component umbenennen, boolean convention "is" & "has" (Google conventions), vielleicht Variablen zusammenfassen mit einem Datentyp, zombie code löschen, längere Methodennamen brauchen. Fileaufbau: 1. Variablen, 2. Constructors, 3. Implementierte Funktionen, 4. Methoden. HTML: mehr in CSS bringen

Präsentation: Titelfolie gestalten, wenn so viel bilder -> Text muss konkreter sein (Konkrete Beispiel von "Basisfunktionen"), soll nicht ablesen, demo war eher ablenkend, Nodes als "lieferanten"

Beim 2. Meeting ist Adrian dabei, wenn er Zeit hat

Scrum Meeting

Die Meiste haben gegen Ende der Iteration weniger gemacht
Marco hat den Code aufgeputzt und Kommentare hinzugefügt

Sprint planning/retrospective

undo/redo, import/export, Menu anfangen zu gestalten wie Layout, Komponente trennen

Demo Organisation

<https://obsproject.com/> - Gratis Video editing software

Produkt demo mit Inputs von Projekt, +-10 minuten video, Inhalt frei (was kann man mit dem Produkt machen, prozess dahinter, einbisschen verkaufend, kein Code)

Alle schreiben ein Ablauf in einem Google Doc

Osterferien Organization

Eigentlich Ferien

Vielleicht am letzte Freitag ein Scrum Meeting

30.03.2021

Scrum Meeting

Zeitplan

Scrum Meeting	~10 min
Kunden Meeting diskutieren	~10 min
Risk Analysis for this week	~5 min
Deliverables for this week	~5 min
Demo: Hints, Probleme, Bemerkungen?	~10 min
Fragen? Bemerkungen?	~5 min

Scrum Meeting

Langsame Start

Kunden Meeting diskutieren

Code nicht gezeigt

Meeting schneller führen

Lawrence: Demo Iteration 1, Korab: Prototype, Jethro: Ziele Iteration 2

Adrian fragen, ob er kommt

Risk Analysis for this week

Workflow könnte man verbessern

Kunde hat andere Erwartungen

Ziel der Demo vielleicht nicht ganz klar

Zeitplanung Demo

Demo: Hints, Probleme, Bemerkungen?

Anfang: Funktionalitäten zeigen (import/export, undo/redo)

Nicht schwierige Wortschatz brauchen

2 Stories präsentieren: Diagram machen, in ihre webseite zeigen

Fragen? Bemerkungen?

Wrapped Text in Node

Branches löschen/umbenennen

30.03.2021

Kunden Meeting

Zeitplan Kundenmeeting

Frage/Aktivität	Zeit
Überblick Projekt (Code Gestaltung, Funktionalitäten, ...)	~10 min
Gewählte Prototype zeigen + diskutieren	~10 min
Fragen? Bemerkungen zur Iteration 1/Entscheidungen?	~5 min
Iteration 2 vorstellen (was zu implementieren, Plan)	~10 min
Fragen? Bemerkungen?	~5 min

Überblick Projekt (Code Gestaltung, Funktionalitäten, ...)

Labels direkt bei Nodes ändern

Knoten sollen automatisch ordnen (Baumstruktur)

Knoten am übergeordnete Knoten herstellen

Node im Zentrum, nicht im Boundary (mit Constraints, vilt mit Animation auseinanderziehen)

Circle dependency Logik vermeiden hinzufügen (Interface auf Knote)

Gravitation von Parentnode hoch (Properties anpassen)

Gewählte Prototype zeigen + diskutieren

Maus zu viel herumwandern

Lieber direkt am Root node (Optionen auf den Rand gezeigt nach Knote druck)

Kontextsensitive/global sensitive Aktionen sollen nicht vermischt sein

Lieber rings herum Pop up Funktion bei Elemente

Interaktiver/integrierter

Farbeset als Kategorisierung

Rechts Kategorie (von uns gewählt)

Rechteck-auswahl (sowie Lasso)

Styles

Iteration 2 vorstellen (was zu implementieren, Plan)

Vielleicht zu viele Stories

Fragen? Bemerkungen?

Träger als der normale Physics

Fixieren ist gut, sowie Root node

Maus über Knote hover -> kommt ein Pin/Schloss, kann man Fixieren aktivieren

Nachbesprechung

Globale Optionen in einer Settings Icon verstecken

Fix x und fix y zusammen/Anchor Node checkbox (appears when hovering over node)

Physics: damping einstellen

Constraints/Relaxation when adding nodes (nodes can't be placed too close to border, too close to other nodes). Far away nodes need to be pulled to graph. Nodes too close need to be pushed away.

Styles als drop down

default/create custom styles

Linke Toolbar nicht wegbringen

Labels sollen real-time im Canvas geändert