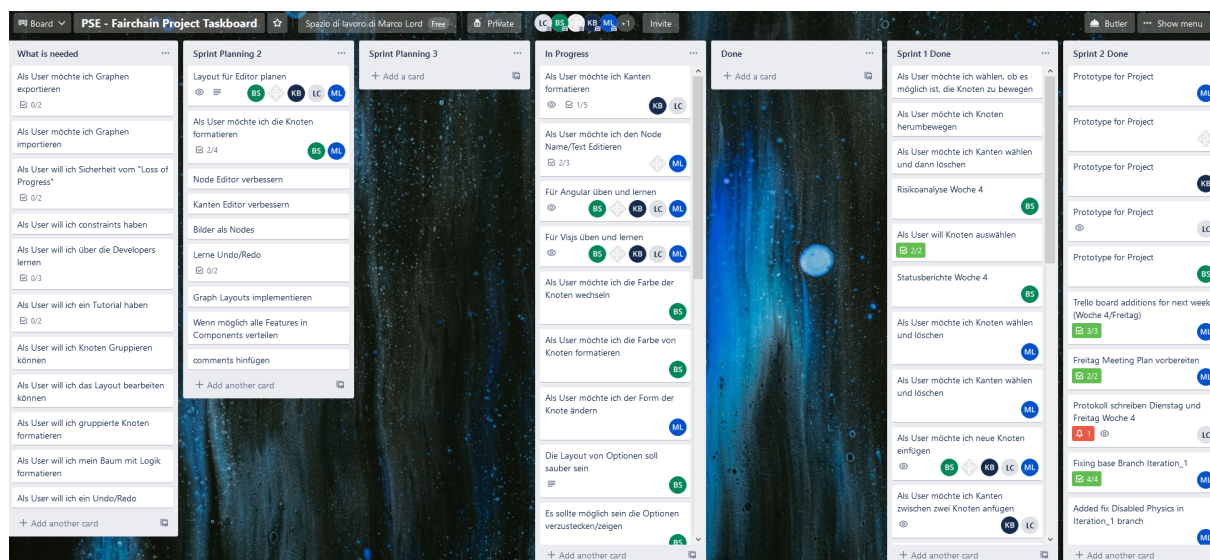


Iteration 1 Analyse

War der Inhalt der Stories nach dem Planning Game klar?

Der Kunde hat uns komplette Freiheit überlassen für die Entwicklung von Stories. Die einzigen Constraints die verlangt wurden, ist das das Projekt unter der Lizenz LGPL 2.1 liegt, es existiert eine Import/Export Funktionalität und dass das Netzwerk übersichtlich sein sollte.

Wir haben somit unsere eigene Stories entwickelt, was zum Vorteil gebracht hat, dass wir uns die Zeit gegeben haben, um uns an die neue Frameworks Vis.js und Angular gewöhnen zu können und uns die Zeit nehmen konnten, die Basis Implementationen zu implementieren.

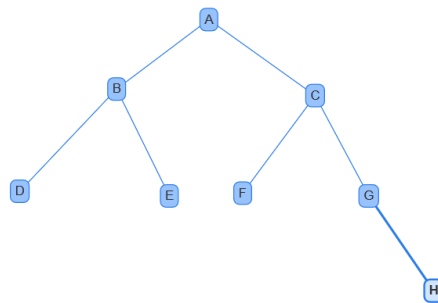


Unsere Taskboard auf Trello

War der Umfang der Stories zu gross/zu klein?

Die Stimmung des Teams ist das die Menge der Stories angemessen war. Das Ziel für die erste Iteration war vor allem das Erlernen von den gebrauchten Technologien: Vis.js und Angular. Mit diesem Wissen versuchten wir, "Basisfunktionen" zu implementieren wie z.B Knoten/Kanten hinzufügen und diese optisch verändern.

Unser selbst gegebene Stories waren genug (nicht zu viele/wenige). Für uns war die Strategie eigentlich uns genug Zeit zu geben, um mit den neuen Frameworks klar zu kommen und die Basisfunktionalitäten zu implementieren.



Ein Screenshot von unseren WebApp nach der 1. Iteration

War die Aufwandschätzung der Stories realistisch?

Bei den eigenen Funktionalitäten (Stories) wurde der Aufwand ein wenig unterschätzt. Beispielsweise die Edit-Node/Edge Funktionalitäten waren komplizierter zu implementieren, da unserer Anfangs Datenstruktur nicht ermöglichte, einzelne Node/Edges zu selektieren und modifizieren. Grundsätzlich konnten wir die Probleme gut Lösen und die Basis Funktionalitäten funktionierend implementieren.

Wurde der Aufwand, sich in neue Programmiersprachen/Technologien einzuarbeiten, realistisch eingeschätzt?

Marco und Jethro mussten sich mehr mühe geben, da sie keine vorkenntnisse auch in Angular hatten (beide haben in ESE Backend entwickelt). Aber dank Blendi, Korab und Lawrence konnten wir eine starke Basiskennntnis in Angular aufbauen. Das Teamwork, Brainstorming und Pair Programming hat uns dabei sehr geholfen.

Es gab grossen Zeitaufwand die Framework Vis.js zu erlernen und richtig anzuwenden, da die Dokumentation veraltet und eher schlecht ist. Natürlich lernen die Teammitglieder weiterhin die Framework, doch der Aufwand für die Programmiersprache und das Framework wurde realistisch eingeschätzt.

Wurde das Entwicklungstempo realistisch eingeschätzt? Gab es Engpässe?

Es gab Engpässe am Anfang und am Ende der ersten Iteration. Diese Engpässe waren kein technologisches Problem, sondern ein organisatorische.

Das Entwicklungstempo des Projektes am Anfang war gebremst, da man mit mangelnder User Stories arbeiten musste. Dies hat uns am Anfang verwirrt, da als Erfahrung auch von ESE uns schon bewusst war, dass wir sicherlich sehr grobe und schlecht erklärte Beschreibungen von Kunden erhalten werden, aber nicht, dass wir überhaupt keine Stories und Grundstruktur des Produktes bekommen wurden (wir mussten uns fast ganz am Bild konzentrieren, die der Kunden im Projektbeschreibung gegeben hat).

Eine gute Trello, Slack organisation und Aufgabenverteilung und ein zweimal wöchentliches Meeting hat uns geholfen, einen guten Entwicklungstempo am Anfang der Iteration 1 zu halten (wir hatten in die erste Woche schon meisten der Basisfunktionalitäten entwickelt)

Das Tempo wurde wieder gebremst am Ende der ersten Iteration, da ein Chaos herrschte im Projekt Repository. Nicht alle Gruppenmitglieder wussten wie man sauber Features mit Git implementierte. Dieses Problem wurde im Team besprochen mit den Assisten Adrian und eine neue Entwicklungsstrategie wird in der zweiten Iteration angewendet.

Kann die gruppeninterne Kommunikation verbessert werden?

Wir haben immer versucht klar und deutlich zu kommunizieren. Wir haben jede Woche 2 Zoom Meetings, wo wir sagen, auf was wir gearbeitet haben und was wir in der Zukunft bearbeiten wollen. Diese Meetings hat uns nicht nur geholfen, uns miteinander zu koordinieren, sondern gab uns auch Zeit, um Lösungen zu besprechen usw.

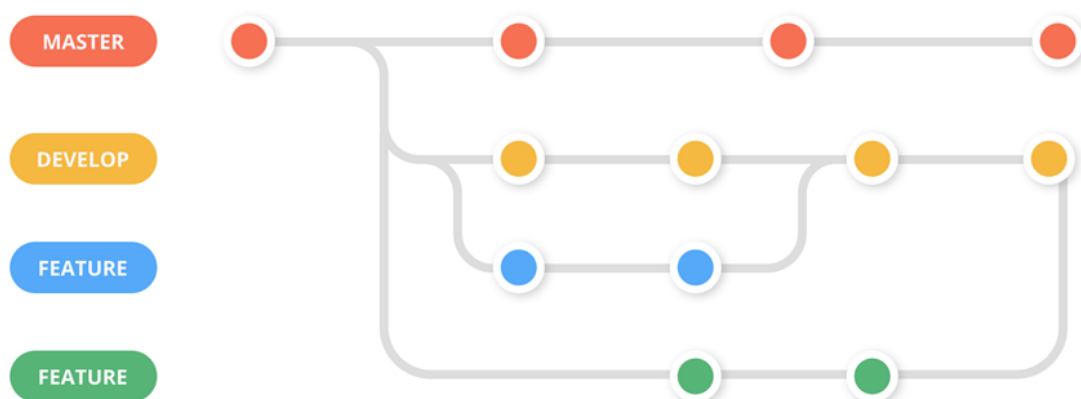
Gegen Ende der ersten Iteration

Die Team-Meetings versuchen wir effizienter zu gestalten anhand einer sehr groben Traktandenliste. Dies gelingt uns zwar schon um einiges Besser, ist aber immer noch ausbaufähig.

Wir hatten zu Beginn Probleme mit der Github Struktur und mit der Effizienz unserer Team-Meeting. Dies versuchen wir mit der Zeit zu verbessern.

In der zweiten Iteration wird eine Entwicklungsstrategie mit GitHub eingesetzt. Dies sollte die chaotische Struktur in Zukunft vermeiden.

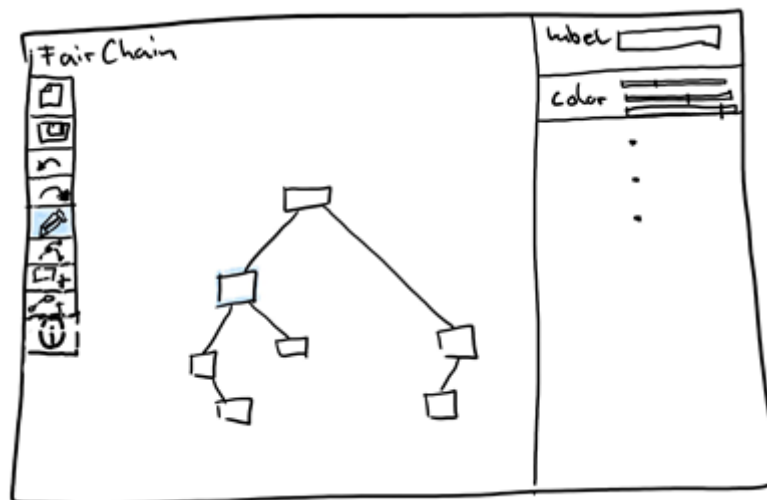
Beim Github einigten wir uns auf eine Struktur und versuchten diese ab sofort so gut es geht einzuhalten.



Der Branchstruktur, für der wir entschieden haben

War die Arbeitsbelastung aller Teammitglieder ähnlich? Sind alle zufrieden?

Da der Product Owner nicht eine offenbare Form hat, war das Team frei ihre eigene Vision für das Projekt zu implementieren. Dadurch war die Arbeitsleistung angemessen. Die Verteilung der Beschäftigungen jedes Mitgliedes wurde in den Meetings besprochen und in den Trello Cards festgehalten. Dadurch empfindet das Team auch, dass Ihre zugeteilte Arbeit gerecht ist. Manchmal einigen wir uns, dass jedes Teammitglied das Gleiche macht, wie z.B beim Designen von Prototypen unserer Applikation.



Der Prototyp, der wir schlussendlich gewählt haben

Gab es "Leerläufe" oder Wartezeiten aufgrund der Abhängigkeiten zwischen den Tasks?

Bis jetzt unsere Tasks waren nicht abhängig miteinander, deswegen gab es keine Wartezeiten. Es gab aber 2 Leerläufe nicht wegen den Tasks, sondern wegen der Organisation. Wir hatten eine kurze Leerlauf beim refactoring vom Code, wo wir probiert haben, aus eine grosse Component viele kleine Components zu machen. Ein weitere partielle Leerlauf, der nicht von Abhängigkeiten zwischen den Tasks verursacht wurde, war, als wir so viele Branches auf Git hatten, dass ein Paar Mitglieder ein bisschen verwirrt waren und nicht mehr wussten, wo Code zu schreiben.

Wieviel Zeit hat jedes Teammitglied investiert für

	Blendi	Marco	Korab	Jethro	Lawrence
Implementation von Stories	32h	36h	20h	15h	20h
Implementation von Testfällen	-	-	-	-	-
Manuelle Tests	5h	8h	6h	4h	4h
Testen, Einarbeiten in neue Technologien	12h	25h	25h	20h	25h
System Administration	8h	16h	4h	3h	7h

System Administration

Uns war ein Systemadministration grundsätzlich nicht nötig, da wir nichts im Server haben. Wir mussten aber Angular installieren und git einrichten. Bei ein Paar Gurppenmitglieder funktionierten diese Systeme nicht so wie sie wollten.

Blendi: ~1h Git-Hub + 5h Statusbericht schreiben + 2h Risikoanalyse

Marco: ~ 6h Git-hub + 10h Trello Organisation

Korab: >1h Git-Hub + 3h Kommunikation mit dem Kunden

Jethro: 3h Git-Hub

Lawrence: ~1h Git-Hub + 6h Protokoll schreiben

Wo ist für die nächste Iteration diesbezüglich der grösste Aufwand zu erwarten?

- Implementation von Stories
 - Die Applikation sollte ein Undo/Redo feature einbauen
 - Es sollte möglich sein ein einfacher Graph bauen zu können
 - Import/Export von Graphen
- Gestaltung der Applikation
 - Die Applikation in Komponente spalten
 - Die Logik der Applikation durch Services implementieren
- Implementation von Testfällen
 - Unit Tests
 - Component Tests

Takeaways vom Team:

Blendi:

- Bei der Präsentation muss man nicht im Detail über die fachspezifisch Dinge sprechen. Man sollte zeigen, was und wie der Team gemacht hat. Man muss sich verkaufen
- Wenn mehrere Personen an derselben Codebasis arbeiten, muss man gute Kommunikation haben
- Bei Meetings muss man einen guten Rhythmus haben

Marco:

- Kritiken bei den Arbeitsablauf konnten uns auf die richtige Entwicklungsstrategie richten (Beispielsweise Pull-requests mit Jethro)
- Teamarbeit war sehr hilfreich für denen die noch nie Angular benutzt hatten und auch um schneller Bugs zu lösen, dies dient auch als Verstärkung unserer Zusammenarbeit.
- Kommunikation ist eine wichtige Schlüssel, um Probleme früh zu finden und eine Lösung zu bearbeiten.
- Gute Arbeitsstruktur dient auch als gute Arbeitsablauf und effizienz.

Korab:

- Wichtig ist sich in der Gruppe auf etwas zu entscheiden, somit kann man Unklarheiten umgehen und schon von Beginn an zu diesem Ziel arbeiten anstatt ohne klares Ziel einfach drauf los arbeiten.
- Die Möglichkeit zu haben bei den kleinsten Unklarheiten die anderen aus der Gruppe zu fragen ist äusserst hilfreich und stärkt die Zusammenarbeit.

Jethro:

- Es ist wichtig sich über alle Abläufe so schnell wie möglich abzusprechen um zukünftige Probleme zu vermeiden (z.B. die organisation von Git).
- Das Team kann schneller und mehr effektiv arbeiten, wenn man seine Kenntnisse in einem klaren Format präsentiert.
- Man sollte Zeit verbringen die minimal benötigte Features zu finden.
- Ein Transkript für Meetings ist sehr wichtig.
- Eine gute Strategie um eine Framework zu lernen (oder andere Technologien), ist durch die Analyse von Beispielen.

Lawrence:

- Eine Gruppenorganisation ohne Product Owner kann sogar positiv auf den Motivation von den Gruppenmitglieder wirken.
- Es ist entscheidend, dass man einen Arbeitsklima erschafft, wo kein Gruppenmitglied Angst hat zu sagen, dass er etwas nicht versteht/ noch nicht kann. Verständnis füreinander haben und die Wille, einander zu helfen hilft da sehr.