Zeitliche Planung, Methodik und Organisation des Projektes Noodle

Zeitliche Planung

Die zeitliche Planung in Aufgabenpakete, Milestones, Projektphasen, etc. haben wir in den entsprechenden Github Repos in Form von Labels, Issues und Milestones dokumentiert und zum Überblick ein Kanban Board im Bereich Projects erstellt, dass alle Informationen schnell und übersichtlich darstellen kann.
Damit der Progress der Tasks mit der Bearbeitung der Issues auch im Kanban Board direkt aktualisiert wird, sind entsprechenden Automationen implementiert.
Zudem sind verschiedene Views gebaut, die helfen sollen sowohl den zeitlichen Kontext als auch den logischen Kontext anzupassen.

Das Kanban Board ist hier zu finden:
Noodle Project

Methodik

In diesem Abschnitt soll präsentiert werden, mit welcher Methodik an dem Projekt gearbeitet werden soll, sowie wieso wir uns für diese entschieden haben.

Vorgehensweise

Als Vorgehensweise haben wir uns für eine Kombination aus der agilen und der traditionellen Vorgehensweise entschieden, da wir zwar an einem großen Projekt arbeiten, aber auch die Vorteile der Projektbasierten Entwicklung nutzen wollen.
Wir nutzen dabei die iterative Vorgehensweise, bei der ein Sprint eine Woche geht und sich jeder zu Beginn des Sprints Gedanken macht, was er schaffen möchte und an dieser Aufgabe dann arbeitet.
Im Laufe der ersten beiden Sprints hat sich dies dazu entwickelt, dass sich jeder, insofern möglich, Aufgaben innerhalb seines Kompetenzbereichs aus dem Stapel des Backlogs nimmt. Sobald an der Spitze des Backlogs keine Aufgaben liegen, die in diesen Bereich fallen, hilft man dann bei den anderen Aufgaben mit.
Um bei der Entwicklung ein direktes Feedback dazu bekommen, wie die letzten Implementierungen das Produkt verändert haben, wird am Ende jedes Sprints ein kompletter Build des Produkts erzeugt und dieser dann entsprechend getestet. Dadurch lässt sich eben auch die kreative Komponente bei der Entwicklung miteinbeziehen, da es durchaus vorkommen wird, dass das Team sich beim Versetzen in die definierten Personas merkt, dass eventuell ein neues Feature benötigt wird oder ein geplantes Feature nicht mehr den Impact bringt, der eigentlich erwartet wurde.
Beim Thema der Dokumentation orientieren wir uns wieder an der planbasierten Methode, sodass während der Entwicklung direkt mit an der Dokumentation des Produktes gearbeitet wird.

Team Meetings

Da wir in der gleichen oder einer sehr ähnlichen Konstellation als Gruppe an mehreren Projekten aus mehreren Fächern gleichzeitig arbeiten, treffen wir uns normalerweise jede Woche Montag, Dienstag und Mittwoch, um gemeinsam an allen Projekten zu arbeiten. Je nachdem in welchem Projekt besonders viele Entscheidungen gemeinsam getroffen werden müssen oder andere Gründe erfordern, dass es am sinnvollsten ist, dass wir gemeinsam daran arbeiten, verändert sich dessen Anteil in der Woche. Generell lässt sich aber sagen, dass der Montagmittag bis Abend meist für gemeinsame Arbeiten an Noodle genutzt wird.
Zusätzlich dazu arbeitet natürlich jeder einzelne entsprechend seiner individuellen Zeitplanung an Noodle weiter und bearbeitet die Aufgabe, die ihm zugeteilt wurde, beziehungsweise die er sich ausgesucht hat. Falls man bei dem Bearbeiten merkt, dass man zum Weiterarbeiten am besten eine Entscheidung des ganzen Teams braucht, kann sich jeder im Gruppen Chat melden und meistens wird dann innerhalb der nächsten 24 Stunden ein außerplanmäßiges Treffen einberufen.

Die planmäßigen Meetings laufen dabei so ab, dass bereits in der Woch davor festgelegt wurde, an welchem der drei Tage wir uns um welches Projekt kümmern würden. Sobald sich ein Meeting um Noodle dreht, läuft dies so ab, dass zu Beginn der regelmäßigen Meetings jeder präsentiert, woran er seit dem letzten Meeting gearbeitet hat und auf welche Probleme er gestoßen ist, bzw. ob er einen Stand hat, den er präsentieren möchte, bzw. diesen mit den anderen abstimmen will.
Während diesem Durchlauf formt sich dann meist schon eine Liste an gemeinsamen ToDo's, die dann nacheinander abgearbeitet werden.
Sobald alle offenen Themen durch sind, wählt jeder dann die Aufgabe für nächste Woche aus, und je nachdem, ob diese Teamaufgaben sind, wird im Anschluss dann an der Aufgabe entweder in der vollen Gruppe oder in den Subgruppen weitergearbeitet. Insofern in dieser Zeit bereits etwas präsentierbares erarbeitet wurde, finden vor Ende des Meetings nochmal alle zusammen, um diesen Stand dann gemeinsam zu besprechen.
Wenn spontane Meetings einberufen wurden, wird der Ablauf entsprechend angepasst und die Person, die eben die Meinung der Gruppe einholen wollte, beginnt damit, ihr Problem bzw. ihren Stand zu präsentieren. Je nachdem, wie wir Zeit haben, wird im Anschluss entweder dann die originale oder eine gekürzte Version des normalen Ablaufs angehängt. Bei gekürztem Ablauf stellt jeder eben kurz seinen Stand vor, bzw. ob er ein Problem hat.

Verwaltung der Backlog-Items / Anforderungen

Aktuelle sind sich alle Anforderungen, die mit der Persona Methode entwickelt wurden, in unserem Zentralen Dokument zur Ideenfindung niedergeschrieben. In den nächsten Wochen sollen dann alle Anforderungen an die V1 von Noodle in konkrete Tasks umgewandelt werden, die dann sowohl in unserem Backlog Dokument als auch als Issue im entsprechenden Repository eingetragen werden.

Zu Beginn von jedem Sprint sucht sich wie oben beschrieben jeder die Einträge aus dem Backlog heraus, an denen er für diesen Sprint arbeitet. Sobald man sich seine Tasks ausgesucht hat, assigned man sich zu den entsprechenden Tasks. Wenn man dann mit der Arbeit an einem Task beginnt, setzt man den Status der Task von "ToDo" auf "In Progress". Wenn man mit der Arbeit an der Task fertig ist, und diese, wenn es Code ist, implementiert ist oder, wenn es Code ist, die anderen Teammitglieder ihr okay gegeben haben, wird der Status der Task auf "Done" gesetzt, damit jeder sehen kann, dass diese Aufgabe erfolgreich abgeschlossen ist.

Der Backlog ist nach Prioritäten sortiert, die sich daran orientieren, wie wichtig die Aufgabe für das Gesamtprojekt ist(Kernfunktionalitäten haben sehr hohe Prioritäten, Gimmicks/ Add-ons eher niedrige) und bis wann die Aufgabe erledigt sein muss.

Organisation

Regeln zur Zusammenarbeit

Aufgrund dessen, dass wir über die letzten Semester bereits sehr gut in diesem Team zusammenarbeiten konnten und es bisher noch zu keinen Problemen innerhalb der Gruppe kam, kann der Regelsatz auf ein Minimum reduziert werden. Zudem haben sich über die Zeit bereits einige Grundlagen eingespielt, die die Zusammenarbeit zudem erleichtern:

- Zusammenarbeit innerhalb der Gruppe:
  Durch die Erfahrungen hat sich der folgende Grundsatz eingespielt:
  “Behandle die Anderen so, wie du auch behandelt werden möchtest.”
  Dies gilt jedoch nicht nur für das grundsätzliche Verhalten bei der Kommunikation, sondern beispielsweise auch für Fälle, dass man jemand natürlich hilft, falls dieser vor einem Problem steht und man selber sich eine Lösung kennt oder schon eine Lösungsidee hat.

- Treffen von Entscheidungen:
  Entscheidungen, die einen große Einfluss auf alle Mitglieder haben, dürfen grundsätzlich nur einstimmig und vollständig getroffen werden, damit die Stimme und Bedenken von jedem gehört werden und sich jeder mit der Entscheidung identifizieren kann.
  Kleinere Entscheidungen dürfen auch innerhalb der Gruppe getroffen werden, die an dem Issue arbeitet. Insofern die Änderung die andere Gruppe aber auch betrifft, beispielsweise bei einer Änderung der Api-Adresse, müssen im nächsten Meeting die Anderen hierüber informiert werden. Falls die Änderung von den anderen Mitgliedern nicht akzeptiert werden sollte, muss die Entscheidung jedoch wieder rückgängig gemacht werden.

- Code Reviews:
  Um eine hohe Qualität des Codes zu gewährleisten und so Bugs in der Anwendung idealerweise schon bei der Produktion zu reduzieren, muss jede Änderung im Code mindestens von einer anderen Person überprüft werden. Hierfür nutzen die eingebauten Funktionen von Github und haben als Regel eingeführt, dass bei Änderungen grundsätzlich ein Merge Request erstellt werden muss und dieser nur von einer anderen Person genehmigt werden darf, die sich in der Anwendung bereits auskennt. Für das Noodle-Backend wäre dies aktuell entweder Max oder Dirk und im Frontend entweder Hannes oder Johannes.
  Wenn die Code-Änderung in Pair-Programming Sessions entstanden ist und einer der Reviewer für die entsprechende Anwendung beteiligt ist, darf die Änderung auch direkt selbst gemergt werden.

Projekt- und Entwicklungs-Guidelines

Zur aktuellen Zeit haben wir folgenden Guidelines festgelegt:

Code Guideline

Da sowohl das Frontend als auch das Backend in der Programmiersprache Javascript geschrieben wird, wurde sich im Team darauf geeinigt, dass die NodeJS Best Practices (zu finden unter https://github.com/goldbergyoni/nodebestpractices) verbindlich eingehalten werden müssen. Um dies zu prüfen nutzen wir es ESLint als Linter.
Als Style-Richtlinien kommt der AirBNB Javascript Style Guide zum Einsatz.
Um Klassennamen, Variablen & Funktionen, sowie Repository Namen sowohl im Code als auch in allen anderen Dokumenten gut voneinander trennen zu können, nutzen wir für die Benennung der Repositorys **snake_case**, für Variablen und Funktionen **camelCase** und für Klassennamen **PascalCase**.

Documentation Guideline

Um einen einheitlichen Stil bei allen Dokumentationen und auch allen Dokumenten zu halten, wurde sich dazu entschieden, dass die gesamte technische Dokumentation auf Englisch erstellt werden soll, da auf der einen Seite Noodle der Weg in andere Länder in der Zukunft offenstehen soll, da dies Teil eines Expansionsplanes ist, und auf der anderen Seite, ein Großteil der technischen Dokumentationen für andere große Software Projekte auch in Englisch gehalten ist.
Die Dokumentation des Codes soll an den entsprechenden Stellen im Code durchgeführt werden, damit der Code möglichst leicht zu lesen und zu verstehen ist. Die Dokumentation hinter der API, Design Ideen, zum gesamten Aufbau oder anderen übergreifen Punkten soll in dem Github-Wiki des entsprechenden Repositories durchgeführt werden.

Alle anderen Dokumente sollen hingegen in Deutsch gehalten werden, da diese Dokumente hauptsächlich für den internen Nutzen oder für eine deutschsprachige Zielgruppe gedacht sind, weshalb hier der Overhead durch eine englische Dokumentenführung vermieden werden sollen.
Dies gilt beispielsweise für den Team-internen Backlog, für die Regeln zur Zusammenarbeit, für die Anforderungsanalyse oder auch für die Ideensammlung.

Tools

Im Rahmen der Projektorganisation werden die folgenden Tools genutzt. Bei den Überlegungen welche Tools wir nutzen möchten, stand im Vordergrund, dass wir möglichst wenig Tools nutzen möchten, die zudem noch gut miteinander harmonieren und mit deren Nutzung wir schon vertraut sind, um so den sonst anfallenden Overhead zu reduzieren.

- Für die Versionskontrolle von Code und Dokumenten nutzen wir Github. Dieses Tool kommt zudem noch zum Automatisieren der Prozesse, sowie zum Aufgaben-Management und zur Projektplanung zum Einsatz. Dies liegt vor allem daran, dass wir durch die verschiedenen Projekte in den letzten Jahren damit bereits sehr gut vertraut sind und zudem jeder von uns Github auch in seiner Freizeit nutzt. Die Entscheidung Github auch für die Projektplanung bzw. für das Aufgaben-Management zu nutzen, kommt daher, dass sich das Aufgaben-Management so direkt in den Coding Cycle integrieren kann.
- Zur schriftlichen Kommunikation innerhalb des Teams, zum Durchführen der virtuellen Teammeetings, sowie zum Vereinbaren der nächsten Treffen wird Discord genutzt, da jeder von uns dieses Tool bereits täglich nutzt.
- Um Dokumente zu erstellen und diese bearbeiten zu können, haben wir uns für Google Docs entschieden, da auch dies wieder ein Tool ist, mit dem wir in den letzten Jahren gute Erfahrungen sammeln konnten. Zudem können wir so problemlos gemeinsam an dem gleichen Dokument arbeiten und sehen dabei die Änderungen der anderen Personen in Echtzeit. Zudem nutzen wir Google Docs noch zum Entwickeln von Ideen, sowie zum Erstellen von kleineren Diagrammen.
- Zum Erstellen von komplexen Diagrammen wird Draw.io eingesetzt, da sich dieses Tool sehr gut in Google Docs und Google Drive integrieren kann und zeitgleich auch eine Versionskontrolle über die Nutzung von Github möglich ist.
- Zum Erstellen von kurzen Sketchups, zur Erstellung des Logos, sowie zum Erstellen von schnellen Mitschriften kommt bei einzelnen Teammitgliedern Goodnotes zum Einsatz. Dies liegt primär daran, dass sich dieses Tool schon in den Uni- und Arbeitsalltag integriert hat und zugleich aktuell auf dem IPad eine der besten Apps zum Anfertigen von Notizen ist. Da hier aber innerhalb der Gruppe keine kollaborativen Arbeiten möglich sind, wird der aktuelle Stand via Screenshots geteilt, um auf diese Weise das Feedback integrieren zu können.