

Projektdokumentation

AUTOR I – E-MAIL

AUTOR II – E-MAIL

HTWK Leipzig

Inhaltsverzeichnis

I	Anforderungsspezifikation	3
I.1	Initiale Kundenvorgaben	3
I.2	Produktvision	3
I.3	Liste der funktionalen Anforderungen	4
I.4	Liste der nicht-funktionalen Anforderungen	4
I.5	Weitere Zuarbeiten zum Produktvisions-Workshop	4
I.6	Liste der Kundengespräche mit Ergebnissen	4
II	Architektur und Entwurf	4
II.1	Zuarbeiten der Teammitglieder	4
II.2	Entscheidungen des Technologieworkshops	4
II.3	Überblick über Architektur	5
II.4	Definierte Schnittstellen	5
II.5	Liste der Architekturentscheidungen	5
III	Prozess- und Implementationsvorgaben	5
III.1	Definition of Done	5
III.2	Coding Style	5
III.3	Zu nutzende Werkzeuge	5
IV	Sprint 1	6
IV.1	Ziel des Sprints	6
IV.2	User-Stories des Sprint-Backlogs	6
IV.3	Liste der durchgeführten Meetings	6
IV.4	Ergebnisse des Planning-Meetings	7
IV.5	Aufgewendete Arbeitszeit pro Person+Arbeitspaket	7
IV.6	Konkrete Code-Qualität im Sprint	7
IV.7	Konkrete Test-Überdeckung im Sprint	8
IV.8	Ergebnisse des Reviews	8
IV.9	Ergebnisse der Retrospektive	8
IV.10	Abschließende Einschätzung des Product-Owners	9
IV.11	Abschließende Einschätzung des Software-Architekten	9
IV.12	Abschließende Einschätzung des Team-Managers	9
V	Sprint 2	10
V.1	Ziel des Sprints	10
V.2	User-Stories des Sprint-Backlogs	10
V.3	Liste der durchgeführten Meetings	10
V.4	Ergebnisse des Planning-Meetings	11
V.5	Aufgewendete Arbeitszeit pro Person+Arbeitspaket	11
V.6	Konkrete Code-Qualität im Sprint	12
V.7	Konkrete Test-Überdeckung im Sprint	12
V.8	Ergebnisse des Reviews	12

V.9	Ergebnisse der Retrospektive	12
V.10	Abschließende Einschätzung des Product-Owners	13
V.11	Abschließende Einschätzung des Software-Architekten	13
V.12	Abschließende Einschätzung des Team-Managers	13
VI	Sprint 3	14
VII	Sprint 4	15
VII.1	Ziel des Sprints	15
VII.2	User-Stories des Sprint-Backlogs	15
VII.3	Liste der durchgeführten Meetings	15
VII.4	Ergebnisse des Planning-Meetings	16
VII.5	Aufgewendete Arbeitszeit pro Person+Arbeitspaket	16
VII.6	Konkrete Code-Qualität im Sprint	17
VII.7	Konkrete Test-Überdeckung im Sprint	18
VII.8	Ergebnisse des Reviews	18
VII.9	Ergebnisse der Retrospektive	19
VII.10	Abschließende Einschätzung des Product-Owners	19
VII.11	Abschließende Einschätzung des Software-Architekten	19
VII.12	Abschließende Einschätzung des Team-Managers	19
VIII	Dokumentation	20
VIII.1	Handbuch	20
VIII.2	Installationsanleitung	21
VIII.3	Software-Lizenz	21
IX	Projektabschluss	22
IX.1	Protokoll der Abnahme und Inbetriebnahme beim Kunden	22
IX.2	Präsentation auf der Messe	22
IX.3	Abschließende Einschätzung durch Product-Owner	22
IX.4	Abschließende Einschätzung durch Software-Architekt	22
IX.5	Abschließende Einschätzung durch Team-Manager	22

I. ANFORDERUNGSSPEZIFIKATION

I.1 Initiale Kundenvorgaben

Autor: xxx

Maecenas sed ultricies felis. Sed imperdiet dictum arcu a egestas. In sapien ante, ultricies quis pellentesque ut, fringilla id sem. Proin justo libero, dapibus consequat auctor at, euismod et erat. Sed ut ipsum erat, iaculis vehicula lorem. Cras non dolor id libero blandit ornare. Pellentesque luctus fermentum eros ut posuere. Suspendisse rutrum suscipit massa sit amet molestie. Donec suscipit lacinia diam, eu posuere libero rutrum sed. Nam blandit lorem sit amet dolor vestibulum in lacinia purus varius. Ut tortor massa, rhoncus ut auctor eget, vestibulum ut justo.

I.2 Produktvision

Autor: Alex Hofmann

Product Vision Board:

Target Group	Needs	Product
-Maschinenbau-Studenten Maschinenbau-Profis -Lehrende	Vgl. zu händisch: einheitlicher, schneller -plattformunabhängig -Open Source -Einfach zu bedienen	-Webanwendung -Als Graph → quasi als Baukasten → Kantengewichtung, Bausteine wählbar -Import/Export von Modellen Normalisierung des Graphen

Die Webanwendung VarG ist entwickelt für Lehrende und Lernende aus dem Maschinenbau Bachelorstudiengang. Diese erleichtert die einheitliche Erstellung, Bearbeitung, Optimierung sowie Import bzw. Exportierung von sogenannten Variantenfolgegraphen. Darunter ist eine graphische Übersicht zu verstehen, die die möglichen Varianten eines Produktionsprozesses für ein Werkstück darstellt.

I.3 Liste der funktionalen Anforderungen

XXX

I.4 Liste der nicht-funktionalen Anforderungen

Autor: xxx

XXX

I.5 Weitere Zuarbeiten zum Produktvisions-Workshop

XXX

I.5.1 Zuarbeit von Autor X

XXX

I.5.2 Zuarbeit von Autor Y

XXX

I.6 Liste der Kundengespräche mit Ergebnissen

Autor: xxx

XXX

II. ARCHITEKTUR UND ENTWURF

II.1 Zuarbeiten der Teammitglieder

XXX

II.2 Entscheidungen des Technologieworkshops

Autor: xxx

XXX

II.3 Überblick über Architektur

Autor: xxx
XXX

II.4 Definierte Schnittstellen

Autor: xxx
XXX

II.5 Liste der Architekturentscheidungen

Autor: xxx
XXX (bewusste und unbewusste Entscheidungen mit zeitlicher Einordnung)

III. PROZESS- UND IMPLEMENTATIONSVORGABEN

III.1 Definition of Done

Autor: xxx
XXX

III.2 Coding Style

Autor: xxx
XXX

III.3 Zu nutzende Werkzeuge

Autor: xxx
XXX

IV. SPRINT 1

IV.1 Ziel des Sprints

Autor: Erik Heldt

Der erste Sprint des VarG-Projekts lief vom 05.12.2019 bis zum 16.12.2019. Ziel war es, eine fundamentale Struktur und grundlegende Funktionalitäten für die Anwendung zu entwickeln, auf denen man später weiter aufbauen kann. Währenddessen konnte man allgemeine Erfahrungen mit dem Ablauf eines Sprints machen.

IV.2 User-Stories des Sprint-Backlogs

Autor: Erik Heldt

Grundstruktur Die Anwendung sollte zu Beginn ein grundlegendes Fundament aufweisen, damit sich alle Teammitglieder vorstellen können, wie am Ende das Programm aussehen soll. Dazu gehörte zu Beginn das Design der Startseite mit dem VarGraph im Zentrum und der Einbindung von Cytoscape in die Programmstruktur.

Datenstruktur für Knoten Es sollte mit Hilfe von Cytoscape herausgefunden werden, wie man Knoten im Programmcode hinzufügen und speichern kann. Dafür sollte dann eine Datei im Programm angelegt werden.

Knoten zu bestehender Datenstruktur hinzufügen Die Anwendung sollte eine einfache Funktionalität zum Erstellen neuer Knoten aka Produktionsschritte erhalten, um sich mit den Cytoscape-Funktionen näher vertraut zu machen. Hier war erstmal noch keine graphische Darstellung in der GUI notwendig, es reichte per Console logs zu testen.

Darstellung eines Graphen in Weboberfläche In der Anwendung sollte zunächst ein statischer Graph mit Hilfe einer Cytoscape-Datenstruktur sichtbar dargestellt werden, damit man sehen konnte, wie so ein „CytoGraph“ überhaupt aussieht. User-Interaktion war hier noch nicht notwendig.

Kanten anlegen Zusätzlich zu Knoten sollten auch Kanten zwischen bestehenden Knoten hinzugefügt werden können. Diese Kanten sollten mit verschiedenen Attributen in der Cytoscape-Datenstruktur gespeichert werden.

Berechnung verschiedener Eigenschaften Anhand der mit den Kanten gespeicherten Attribute sollte eine Funktionalität entwickelt werden, welche die Gesamtkosten (Auswahl von Geld oder Zeit) aller unterschiedlichen Pfade berechnen und anzeigen sollte. Dies war der erste Schritt in Richtung Optimierung, d.h. später sollte diese Funktionalität automatisch den günstigsten Pfad herausfinden und anzeigen.

IV.3 Liste der durchgeführten Meetings

Autor: Erik Heldt

- Planning - 05.12.2019
- Weekly Scrum 1 - 09.12.2019
- Weekly Scrum 2 - 12.12.2019
- Review - 16.12.2019
- Retrospektive - 19.12.2019

IV.4 Ergebnisse des Planning-Meetings

Autor: Erik Heldt

Im Planning-Meeting erklärten die Projektmanager zu Beginn noch einmal kurz, wie ein Sprint im Allgemeinen abläuft und haben auf die Bedeutsamkeit der Coding Guidelines hingewiesen. Anschließend wurden die ersten User-Stories vom Project Owner vorgestellt und von den Bachelorstudenten per Finger-System in ihrer Komplexität eingeschätzt. Weiterhin wurde festgelegt, dass die Bachelorstudenten während des Sprints die User-Stories selbst in Tasks aufteilen und diese dann bearbeiten sollen.

IV.5 Aufgewendete Arbeitszeit pro Person+Arbeitspaket

Autor: xxx

Arbeitspaket	Person	Start	Ende	h	Artefakt
Vue.js "Getting Started" Tutorial durcharbeiten (für alle)	Buchmann, Lennart	07.12.19	07.12.19	3	Tutorial abgeschlossen
Beispielgraph erstellen	Buxel, Nils	09.12.19	09.12.19	1	index.js
Kürzesten Weg mit A*-Algorithm berechnen u anzeigen lassen	Buxel, Nils	16.12.19	16.12.19	1	index.js
Funktionen zu Buttons hinzufügen	Gwozdz, Jonas	14.12.19	16.12.19	4	MenuControls.vue
Task: Einbindung in Vue-Dateistruktur	Heldt, Erik	15.12.19	15.12.19	3	MenuControls.vue, BasicData.js
Graphenanordnung	Heldt, Erik	05.12.19	05.12.19	3	Graphenanordnung.pdf
Vue.js "Getting Started" Tutorial durcharbeiten (für alle)	Heldt, Erik	11.12.19	11.12.19	2	Tutorial abgeschlossen
Funktionen zu Buttons hinzufügen	Henning, Tim	10.12.19	10.12.19	2	MenuControls.vue
Vue.js "Getting Started" Tutorial durcharbeiten (für alle)	Henning, Tim	06.12.19	06.12.19	3	Tutorial abgeschlossen
Einbindung von Cytoscape in Vue	Herterich, Linus	10.12.19	10.12.19	4	index.js
Buttons für Knoten und Kantenerstellung	Herterich, Linus	13.12.19	13.12.19	3	CreateControls.vue
Knoten zu Graph hinzufügen	Herterich, Linus	16.12.19	16.12.19	2,5	index.js, CreateControls.vue
Grundstruktur aufbauen	Herterich, Linus	05.12.19	07.12.19	9,5	Vue-Dateistruktur, sämtliche Startkom
Task: Basic Datenstruktur	Hohlfeld, Julius	15.12.19	15.12.19	8	BasicData.js, MenuControls.vue

IV.6 Konkrete Code-Qualität im Sprint

Autor: Erik Heldt

Zu Beginn wurde viel experimentiert und hauptsächlich sollte der Code erstmal ein funktionierendes Programm erzeugen, weswegen weniger auf die Qualität geachtet wurde. Trotzdem wurde sich größtenteils an die Coding Conventions gehalten und bereits einige Kommentare verfasst.

IV.7 Konkrete Test-Überdeckung im Sprint

Autor: Erik Heldt

Da der erste Sprint größtenteils nur zur Erstellung einer grundlegenden Datenstruktur und zur Einarbeitung in JavaScript und den genutzten Frameworks bzw. Bibliotheken gedient hat, gab es noch keine Tests.

IV.8 Ergebnisse des Reviews

Autor: Erik Heldt

Im ersten Review-Meeting stellten die Bachelorstudenten ihre Ergebnisse aus dem Sprint vor und die Manager gaben ihr Feedback dazu. Da sich die meisten Teammitglieder noch nicht richtig in Vue.js und Cytoscape einarbeiten konnten und teilweise große Schwierigkeiten mit den Frameworks hatten, gab es noch viele offene Aufgaben und nicht jeder hatte etwas vorzuzeigen. Als erstes stellten Julius H. und Erik die Datenstruktur für die Knoten vor. Weiterhin zeigte Julius, wie ein Knoten in der Anwendung dargestellt wird und dass dieser durch ungeschickte Verschiebung und Skalierung aus der GUI verschwinden kann. Deshalb kamen Vorschläge, zukünftig den Zoom zu limitieren und das grundsätzliche Graph-Layout nochmal zu überarbeiten. Um allen den Einstieg in die neuen Programmiersprachen und Bibliotheken etwas zu vereinfachen, stellte daraufhin Linus die Grundstruktur vor und erklärte noch einmal genau die einzelnen Elemente in der Dateistruktur. Weiterhin zeigte er, wie man ESLint-Fehler bei der Konsolenausgabe verhindern kann. Danach wurde zwischen den Managern und den Bachelorstudenten noch die zukünftige Berechnung der kürzesten Wege und die unbearbeiteten User-Stories besprochen und dass diese in den nächsten Sprint mit einfließen werden. Zum Schluss wurden noch ein paar allgemeine Fragen zum Testen und zu Git geklärt.

IV.9 Ergebnisse der Retrospektive

Autor: Erik Heldt

In der Retrospektive konnte jedes Teammitglied vor an die Tafel gehen und verschiedene Aspekte des Sprints mit einem Strich in einer Tabelle bewerten. Die Bewertung ging ausgeglichen aus. Die Gruppenleistung und das Gesamtergebnis waren gut, aber die Einzelleistungen der meisten Teammitglieder nicht. Viele Aufgaben blieben offen und wurden nicht erledigt, wozu in der Diskussion verschiedene Gründe angeführt wurden. Einerseits war es für die meisten schwer, sich selbst in die neue Programmierungsumgebung samt den Frameworks und Bibliotheken einzuarbeiten. Andererseits wussten viele nicht, was und wie viel sie machen sollten, was auf die nicht festgelegte Aufgabenteilung im Planning und die schlechte Kommunikation im Team während des Sprints zurückgeführt wurde. Letzteres Problem plante man damit zu lösen, in zukünftigen Plannings immer direkt Verantwortliche für bestimmte User-Stories festzulegen und entsprechende Tickets sofort im Anschluss zu erstellen und zuzuweisen. Beim Thema der Daily Meetings ist man zu dem Schluss gekommen, dass diese wenn möglich immer persönlich bleiben sollten und nur in Ausnahmefällen online z.B. über Discord stattfinden sollten. Weiterhin wurde diskutiert, ob die Zeitspanne zwischen Donnerstag und Montag evtl. zu kurz ist, um schon weitreichende Ergebnisse zu erzielen, da am Wochenende einige Teammitglieder nicht programmieren können. Deshalb sollten die ersten Meetings beim nächsten Sprint stattdessen Montag und Donnerstag stattfinden. Ein weiterer Themenpunkt war die Organisation im Git. Es wurde festgelegt, dass der Master-Branch während des Sprints unberührt bleiben sollte, da dieser immer lauffähig sein muss. Stattdessen sollte sich jeder seinen eigenen Branch erstellen und diesen nach Abschluss der eigenen Aufgaben auf den neuen Developer-Branch namens "targetbranch" mergen. Am Ende jedes Sprints würde dann der Developer-Branch mit dem Master-Branch gemerged werden.

IV.10 Abschließende Einschätzung des Product-Owners

Autor: xxx

XXX

IV.11 Abschließende Einschätzung des Software-Architekten

Autor: xxx

XXX

IV.12 Abschließende Einschätzung des Team-Managers

Autor: xxx

XXX

V. SPRINT 2

V.1 Ziel des Sprints

Autor: Linus Herterich

Nachdem im ersten Sprint hauptsächlich die Grundstruktur sowie erste Datenstrukturen entworfen wurden, war es nun wichtig, dass sich das gesamte Team im Sprint 2 mit der Projektstruktur (besonders mit dem Framework Vue) auseinandersetzt und erste UserStories direkt am Code umsetzt. Zudem blieben einige Tickets noch vom letzten Sprint offen, welche nun auch bearbeitet werden sollten.

V.2 User-Stories des Sprint-Backlogs

Autor: Linus Herterich

- **Designumsetzung nach Adobe Preview**
Als Benutzer der WebApplikation möchte ich ein ansehnliche und intuitive Oberflächengestaltung haben, damit ich die Applikation gerne verwende.
- **Authentifizierung eines Nutzers**
Als Nutzer möchte ich mich in die Web Applikation einloggen können, damit nicht jeder meine erzeugten Graphen einsehen kann.
- **Logische verknüpfung zwischen Knoten erstellen**
(wurde in Sprint 1 nicht abgeschlossen)
Ein Nutzer muss eine Abfolge der Knoten definieren können, damit ersichtlich wird welcher Produktionsschritt auf den nächsten folgt
- **Berechnung der Eigenschaften des Gesamtgraphs**
(wurde in Sprint 1 nicht abgeschlossen)
Ein Nutzer der Webanwendung VarG muss die berechneten gesamt Eigenschaften jedes zusammenhängendes Pfades ausgeben lassen können um eine Auswahl eines Pfades zu treffen.
- **Datenstruktur Ausarbeiten & Knoten zu einer vorhandenen Datenstruktur hinzufügen**
(wurde in Sprint 1 nicht abgeschlossen)
Als Nutzer möchte ich Knoten zu der Datenstruktur hinzufügen können um die möglichen Produktionsschritte des Werkstücks überblicken zu können

V.3 Liste der durchgeführten Meetings

Autor: Linus Herterich

- 19.12.2019: Planning Meeting
- 23.12.2019: Daily Meeting (in Discord)
- 28.12.2019: Daily Meeting (in Discord)
- 05.01.2020: Review Meeting
- 06.01.2020: Retrospektive

V.4 Ergebnisse des Planning-Meetings

Autor: Linus Herterich

Neben der Aufgabenverteilung wurde im Planning darüber gesprochen, dass die Arbeitsaufteilung im letzten Sprint nicht gut geklappt hat. Es wurde anschließend beschlossen im nächsten Sprint die User-Stories direkt an Studenten zuzuweisen, damit jeder einen Teilbereich hat, den er bearbeiten muss.

Desweiteren wurde eine Änderung im Git angekündigt. In Zukunft müsse der "Master"-Branch während eines Sprints immer gleich bleiben und Funktionalitäten werden auf einen "Developer"-Branch gemerged. Am Ende des Sprints wird dann der "Developer"-Branch auf den "Master"-Branch gemerged. wichtig ist, dass der "Master"-Branch zu jedem Zeitpunkt lauffähig ist.

Für den folgenden Sprint wurde beschlossen, die Daily Meetings online (auf einem Discord Server) abzuhalten, da viele Studenten über die Weihnachtsferien in der Heimat sind und somit ein persönliches wöchentliches treffen nicht möglich wäre.

V.5 Aufgewendete Arbeitszeit pro Person+Arbeitspaket

Autor: Linus Herterich

Arbeitspaket	Person	Start	Ende	h	Artefakt
UI: Login	Berger, Matthias	22.12.19	22.12.19	3,5	Login Funktionalität & Design
UI: Login	Buchmann, Lennart	22.12.19	22.12.19	6	Login Funktionalität & Design
UI: Grapheneditor	Gwozdz, Jonas	23.12.19	04.01.20	9	GraphHeader.vue, Toolbar.vue
Task: Einbindung in Vue-Dateistruktur	Heldt, Erik	19.12.19	19.12.19	0,25	BasicData.js
Abrufbaren Knoten in Graph einfügen	Heldt, Erik	23.12.19	26.12.19	3,5	BasicData.js, TestDatabase.js
Testdatenbank mit Speichern und Laden	Heldt, Erik	27.12.19	27.12.19	3,5	TestDatabase.js
Highlighting eines kürzesten Pfades nach Anwendung des A* Algorithmus	Henning, Tim	24.12.19	03.01.20	9	OptimizeControls.vue, index.js -> Graph Highlighting
Protokoll: Meeting 19.12.19	Herterich, Linus	19.12.19	19.12.19	1	meeting_19_12_19.pdf
UI: Login	Herterich, Linus	20.12.19	20.12.19	5	LoginForm.vue, Login.vue
UI: Home	Herterich, Linus	23.12.19	23.12.19	7	HomeMenu.vue (component), Home.vue (view), Menu.vue (view)
UI: Neuer Graph	Herterich, Linus	28.12.19	28.12.19	1,5	NewGraph.vue (view), NewGraph.vue (component)
UI: Grapheneditor	Herterich, Linus	02.01.20	04.01.20	11,75	Graph.vue (view), zahlreiche components
Graph zu Datenstruktur hinzufügen	Hohlfeld, Julius	21.12.19	23.12.19	4	BasicData.js, TestDatabase.js

Testdatenbank mit Speichern und Laden	Hohlfeld, Julius	27.12.19	03.01.20	8	BasicData.js, Test-Database.js, index.js, JSonPersistence.js
Mergen und Anpassen	Hohlfeld, Julius	04.01.20	04.01.20	2	Bugs entfernt & Mergekonflikte behoben
UI: Datenbank-Import Fenster	Karkoutli, Alaa Aldin	31.01.20	04.01.20	12,5	Database.vue (view), DatabaseForm.vue (component)
Kanten zu Graph hinzufügen	Koch, David	23.12.20	04.01.20	10	Änderungen an index.js, CreateControls.vue (component)

V.6 Konkrete Code-Qualität im Sprint

Autor: Linus Herterich

Es wurde sich größtenteils an die Coding-Guidelines gehalten. An wichtigen Stellen sowie vor jeder Funktion wurden Kommentare geschrieben. Die Trennung zwischen Views und Components sowie die Auslagerung der Style-Dateien wurde ebenfalls eingehalten.

V.7 Konkrete Test-Überdeckung im Sprint

Autor: Linus Herterich

Ein Student wurde beauftragt bis zum Ende des Sprints ein geeignetes Test-Framework zu finden. Somit wurden während des Sprints noch keine Tests geschrieben.

V.8 Ergebnisse des Reviews

Autor: Linus Herterich

Es wurden fast alle UserStories umgesetzt. Somit war der zweite Sprint erfolgreich. Alle Studenten konnten sich in das Projekt einarbeiten und haben die Strukturierung größtenteils verstanden und eingehalten.

Das User-Interface wurde nach der Designvorlage umgesetzt und die ersten Graphen-Funktionen (Hinzufügen von Knoten und Kanten & Optimieren) funktionieren bereits.

Da noch nicht feststeht, wo die Software gehostet werden soll und wie die Datenbank-Funktionalität umgesetzt werden soll, wurde zunächst eine lokale Speicherlösung als "Datenbank" verwendet. Somit konnten die Speichern- und Laden-Funktionen erfolgreich implementiert werden.

Die Login-Funktionalität ist derzeit nur sporadisch eingerichtet und wird finalisiert, sobald feststeht, wie die Authentifizierung der Nutzer erfolgen soll (Anbindung an HTWK Login?).

Leider ist immernoch kein geeignetes Testframework gefunden worden, mit dem sich sowohl Vue.js als auch cytoscape (Graphen-Funktionalitäten) testen lassen.

V.9 Ergebnisse der Retrospektive

Autor: Linus Herterich

Das Happiness-Barometer für diesen Sprint ist sehr gut ausgefallen. Das liegt hauptsächlich an der guten Aufgabenverteilung sowie an den großen Erfolgen, die diesen Sprint erzielt wurden.

Kritisiert wurde die Kommunikation gegen Ende des Sprints. Das finale Mergen aller Branches war zu hektisch und unsicher.

Es wurde sich darauf geeinigt in Zukunft zwei Dailies pro Woche abzuhalten und das letzte Meeting eines Sprints zum gemeinsamen Mergen zu verwenden.

V.10 Abschließende Einschätzung des Product-Owners

Autor: xxx

XXX

V.11 Abschließende Einschätzung des Software-Architekten

Autor: xxx

XXX

V.12 Abschließende Einschätzung des Team-Managers

Autor: xxx

XXX

VI. SPRINT 3

VII. SPRINT 4

VII.1 Ziel des Sprints

Autor: Jonas Gwozdz

Während der Semesterferien haben wir an Sprint 4 weitergearbeitet. Dieser dauerte vom 23.01.2020 bis zum 09.04.2020. Der Ablauf war dabei weitestgehend planmäßig, bis auf dass die Meetings zum Review und der Retrospektive wegen Corona ohne persönliches Treffen stattfinden mussten. In der Vorlesungsfreien Zeit besprachen wir uns gelegentlich über den aktuellen Zwischenstand. Der größte Fortschritt am Projekt wurde während der letzten beiden Wochen erzielt.

VII.2 User-Stories des Sprint-Backlogs

Autor: Jonas Gwozdz

- **Tests für bereits geschriebenen Code**
Als Benutzer möchte ich eine Software benutzen, die getestet ist, damit keine unerwarteten Probleme auftauchen.
- **Validierung der möglichen Eingaben**
Als Nutzer möchte ich bei versehentlicher falscher Eingabe wenn möglich gewarnt werden, damit ich nichts falsches abspeichere.
- **Bug: Validation bei gleichem Knoten-Namen**
- **Darstellung von Kanten/Attributen**
Als Benutzer will ich alle Kanten/Knoten gleichzeitig sehen können(nicht übereinander), damit ich einen schnelleren Überblick über das gesamte Konstrukt bekomme.
- **Bug: Mehrere Edges zwischen Knoten nicht möglich**
Wenn man mehrere Kanten zwischen zwei Knoten anlegt, sind diese nicht sichtbar. Löscht man dann einen Knoten, an dem diese unsichtbaren"Knoten hängen, so stürzt cytoscape ab.
- **Remodel von Component NewGraph**

VII.3 Liste der durchgeführten Meetings

Autor: Jonas Gwozdz

- 23.01.2020: Planning
- 05.03.2020: Weekly
- 12.03.2020: Weekly
- 06.04.2020: Review
- 09.04.2020: Retro

VII.4 Ergebnisse des Planning-Meetings

Autor: Jonas Gwozdz

Anwesend: Alex, Julius J., Julius H., Linus, Jonas, Erik, Lennart, Nils, Tim, David, Matthias, Manuel

Innerhalb dieses Meetings haben wir die Schwerpunkte des Sprints festgelegt und über den Workload über die Vorlesungsfreie Zeit diskutiert und den Zeitaufwand der User-Stories abgeschätzt.

oberste Priorität: Tests

Da wird bis zum bisherigen Zeitpunkt keine Testumgebung gefunden haben, die sich auf unseren Cytoscape-Graphen anwenden lässt, und wir dadurch viel Nachholbedarf in Sachen Testen hatten, musste dieses Ticket am dringendsten abgearbeitet werden.

Sprint über Semesterferien

Wir haben uns im Planning darauf geeinigt, den Sprint über die Semesterferien mit weniger User-Stories als üblich auszulegen, da nicht alle Teammitglieder in dieser Zeit voll verfügbar waren, Grund dafür waren vor Allem die noch andauernden Prüfungen und die anschließenden Ferien, die evtl. schon anderweitig verplant waren. Zudem haben wir uns darauf geeinigt, regelmäßig Absprache über den Fortschritt unserer Arbeit zu halten.

Datenbanken

Die Datenbankrecherche hat ergeben, dass für unsere Zwecke MySQL oder NodeJS am optimalsten wäre. Die Definition der Datenbankschnittstelle zwischen DB und Frontend muss ebenfalls noch erledigt werden. Zudem haben wir festgestellt, dass die bisher entworfene Datenbankoberfläche optisch nicht zum Rest der Anwendung passt, und deshalb überarbeitet werden muss.

Weitere Sprintziele:

- Optimierung der Kostendarstellung
- negative Zahleingaben abfangen
- automatische Zoomfunktion bei Knoten- oder Kantenwahl
- allgemeine Bugfixes

VII.5 Aufgewendete Arbeitszeit pro Person+Arbeitspaket

Autor: Jonas Gwozdz

Arbeitspaket	Person	Start	Ende	h	Artefakt
Tests für bereits geschriebenen Code	Heldt, Erik	04.03.20	04.03.20	2	Tests für ModifyData-Controls.vue
Neue Strukturierung	Heldt, Erik	26.01.20	26.01.20	1	Umstrukturierung des Projekts
Header Buttons und Metadaten-Speicherung	Heldt, Erik	05.03.20	12.03.20	6,75	GraphHeader.vue
Aufräumen der Branches im GitLab	Heldt, Erik	29.03.20	29.03.20	1	Organisatorische Aufgabe
Entfernen veralteter Komponenten und Methoden	Heldt, Erik	31.03.20	31.03.20	2	Organisatorische Aufgabe

Tests für Graphoptimierung	Henning, Tim	04.04.20	40.40.20	12	vargraph.spec.js
Tests für bereits geschriebenen Code	Herterich, Linus	30.01.20	12.02.20	7,5	/code/cypress/integration/...
Header Buttons und Metadaten-Speicherung	Herterich, Linus	28.03.20	31.03.20	2,25	/vargraph/graph/... & GraphHeader.vue
Aufräumen der Branches im GitLab	Herterich, Linus	30.03.20	30.03.20	1	Organisatorische Aufgabe
Darstellung von Kanten/Attributen	Herterich, Linus	03.04.20	03.04.20	2	VarGraph.vue
Remodel von Component NewGraph	Herterich, Linus	30.03.20	30.03.20	3	/vargraph/graph/...
Refactoring	Herterich, Linus	29.03.20	30.03.20	9	/vargraph/graph/...
Validierung: Login	Herterich, Linus	31.03.20	30.03.20	1,5	/components/login/LoginForm
Einheitliche Alerts	Herterich, Linus	31.03.20	31.03.20	3	Dialogs.vue
Validierung CreateControls & DetailControls	Herterich, Linus	31.03.20	01.04.20	5,5	CreateControls.vue & DetailControls.vue
Bug: Mehrere Edges zwischen Knoten nicht möglich	Herterich, Linus	01.04.20	01.04.20	2	/vargraph/graph/...
Knoten dort erstellen, wo rechtsklick passiert	Herterich, Linus	01.04.20	01.04.20	1,5	/vargraph/graph/...
keybinds für Menüs	Herterich, Linus	02.04.20	02.04.20	1	
Keine Knoten aufeinander schieben	Herterich, Linus	02.04.20	02.04.20	3	/vargraph/graph/...
Einstellungsmenü erstellen	Herterich, Linus	03.40.20	05.04.20	5,5	
Tests für bereits geschriebenen Code	Hohlfeld, Julius	05.02.20	04.03.20	10	ZoomControls.spec & SaveMenu.spec & NewGraphMenu.spec & DownloadMenu.spec
Dialogfenster für Speichern, Laden und Export	Hohlfeld, Julius	24.01.20	24.01.20	2	Toolbar.vue
Validierung der möglichen Eingaben	Hohlfeld, Julius	06.04.20	06.04.20	2	divers
Refactoring	Hohlfeld, Julius	31.03.20	31.03.20	2	/vargraph/graph/...
Testing für Kanten hinzufügen	Koch, David	22.03.20	02.04.20	5	addEdges.spec

VII.6 Konkrete Code-Qualität im Sprint

Autor: Jonas Gwozdz

Die Codequalität im allgemeinen wurde während des Sprints erheblich durch das Refactoring verbessert. Zudem wurden in nahezu allen Dateien einleitende Kommentare geschrieben, um die

zukünftige Identifizierung der gebrauchten Dateien schneller und übersichtlicher zu gestalten.

VII.7 Konkrete Test-Überdeckung im Sprint

Autor: Jonas Gwozdz

Die geschriebenen Cypress-Tests decken bereits eine Vielzahl an Funktionalitäten des Programms ab. Dazu zählen die Buttons für die Database, den Download, das Ausloggen. Zudem wurde getestet: der Speicherdialog, die Zoomeinstellungen, der Header des Graphen, das Hinzufügen von Knoten und das Erstellen eines neuen Graphen.

VII.8 Ergebnisse des Reviews

Autor: Jonas Gwozdz

Anwesend: David, Erik, Julius J., Julius H., Jonas, Linus, Manuel, Matthias, Tim

Im Rahmen des Reviews haben wir wie gewohnt die Ergebnisse des Sprint bewertet und Schwierigkeiten besprochen.

generelle Schwierigkeit: Testen

Um unsere Programm zu testen, entschieden wir uns für das Framework "Cypress"entschieden. dieses bietet End-to-End Testing an, welches allerdings nur Ausgaben des Programms auswerten kann, und deshalb sozusagen keinen Blick unter die Haube zulässt, und somit eventuell Fehler unentdeckt bleiben.

David:

- Tests für Knotenfunktionalität geschrieben
- mit Kantentests begonnen

Erik:

- Data Controls durch Header Buttons ersetzt
- Editierungsfenster entfernt
- Header Buttons getestet

Jonas:

- Testübersicht erstellt
- Möglichkeit zum Informationsaustausch über Lücken und Bugs in Tests bereitgestellt

Julius H.:

- Tests für Toolbar, Zoom-Controls, Buttons und Eingabereihenfolgen geschrieben

Julius H, Erik, Linus:

- Refactoring des Graphen, Bugfixing und Validierung von Eingaben

Linus:

- Dialogue-Popups erstellt
- Kürzelgenerierung implementiert

- Knotenüberlagerung unterbunden, Mindestabstand implementiert
- Einstellungsmenü erstellt und Implementation begonnen
- Recherche zu Datenbankfenster

VII.9 Ergebnisse der Retrospektive

Autor: Jonas Gwozdz

Anwesend: Alex, Erik, Julius J., Julius H., Jonas, Linus, Matthias, Tim

Zu Beginn des Sprints gab es keine Fortschritte zu vermelden, da vorerst die Prüfungen zu überstehen waren. In den beiden Wochen vor Sprintende wurden allerdings die wichtigsten User-Stories und sogar etwas mehr abgearbeitet.

Positiv	Negativ
-produktive Endphase -viel Motivation bei Einigen	-anfangs keine Kommunikation - wenig Motivation bei Einigen -vereinzelt Tests ohne Sinn -ausgefallene Meetings

VII.10 Abschließende Einschätzung des Product-Owners

Autor: xxx

XXX

VII.11 Abschließende Einschätzung des Software-Architekten

Autor: xxx

XXX

VII.12 Abschließende Einschätzung des Team-Managers

Autor: xxx

XXX

VIII. DOKUMENTATION

VIII.1 Handbuch

Autor: David Koch

- **Login**

Beim Login ist der HTWK-Login (wie Opal, Qis, etc.) zu verwenden. (Anbindung an Shiboleth wurde leider noch nicht umgesetzt. Derzeitiger Login mit Username: VarG, Passwort: 2020)

- **Neuen Graphen erstellen**

Über den Menüpunkt 'Neuen Graphen erstellen' im Startfenster gelangt man in das 'Neues Produkt' Fenster. Hier kann Name und Stückzahl des neuen Graphen festgelegt werden. Über den 'Starten' Button wird der neue Graph erstellt. Im Graphenfenster gelangt man über den 'Neuer Graph' Button zurück ins Startmenü. Über die Stift-Icons neben dem Produktnamen und der Stückzahl können diese nachträglich bearbeitet werden.

- **Erstellen und Bearbeiten von Zustände**

Das 'Neues Teil' Menü öffnet sich entweder über Rechtsklick innerhalb des Graphenfensters, gefolgt von einem Linksklick auf 'Neues Teil' oder durch einen Linksklick auf den Plus-Button in der rechten unteren Ecke, gefolgt von einem Linksklick auf 'Neues Teil'. Hier kann neben Name, Kürzel und Farbe des Zustands auch ein dazugehöriges Icon mittels URL gewählt werden. Alle Angaben außer dem Icon sind Pflichtangaben, sodass der 'erstellen' Button erst betätigt werden kann, wenn alle diese Felder beschrieben sind. Mit einem Linksklick auf einen bereits erstellten Zustand öffnet sich das 'Teil bearbeiten' Menü. Dieses ist aufgebaut wie das 'Neues Teil' Menü. Hier können alle Eigenschaften des angeklickten Zustandes bearbeitet werden.

- **Erstellen und Bearbeiten von Bearbeitungsschritten**

Das 'Neuer Bearbeitungsschritt' Menü öffnet sich entweder über Rechtsklick innerhalb des Graphenfensters, gefolgt von einem Linksklick auf 'Neuer Bearbeitungsschritt', oder durch einen Linksklick auf den Plus-Button in der rechten unteren Ecke, gefolgt von einem Linksklick auf 'Neuer Bearbeitungsschritt'. Im ersten Teil können Name, Kürzel sowie Start- und Endzustand gewählt werden. Im zweiten Teil definiert man Losgröße, Zeit- und Geldkosten sowie Zeit und Geldrüstkosten des Bearbeitungsschrittes. Alle Felder sind Pflichtfelder, es müssen also alle Felder beschrieben sein, um den 'erstellen' Button betätigen zu können. Des Weiteren kann man Bearbeitungsschritte auch mittels drag'n'drop erstellen. Dafür bewegt man den Mauszeiger über den Zustand, der als Startzustand dienen soll, klickt auf das neuer Bearbeitungsschritt-Icon und zieht die entstehende Linie zu dem Endzustand. Anschließend öffnet sich der zweite Teil des 'Neuer Bearbeitungsschritt' Menüs, in dem dann die übrigen Eigenschaften nachträglich eingefügt werden müssen. Mit einem Linksklick auf einen bereits erstellten Bearbeitungsschritt öffnet sich das 'Bearbeitungsschritt bearbeiten' Menü. Dieses ist aufgebaut wie das 'Neuer Bearbeitungsschritt' Menü. Hier können alle Eigenschaften des angeklickten Bearbeitungsschritt geändert werden.

- **Graph optimieren**

Um den besten Weg von einem der Startzustände zum Endzustand zu finden, klickt man auf 'Graph optimieren' unter 'Gesamtkosten' (um nach Kosten zu optimieren) oder unter 'Gesamtzeit' (um nach Zeit zu optimieren), Start- und Endzustände werden in diesem Fall automatisch gewählt. Nach dem Betätigen des Buttons verschwindet dieser und wird ersetzt durch die Gesamtkosten bzw. die Gesamtzeit des besten Weges im Graphen, außerdem ist dieser Weg Orange markiert. Abhängig davon, ob nach Kosten oder Zeit optimiert wurde wird

das entsprechende Icon sowie das Wort ('Gesamtkosten' oder 'Gesamtzeit') ebenfalls Orange markiert. Möchte man die Parameter der Optimierung eigenhändig bearbeiten, lässt sich das entsprechende Menü öffnen, indem man entweder auf eines der Zahnräder (neben 'Gesamtkosten' oder 'Gesamtzeit') klickt oder auf 'Einstellungen' am oberen Rand klickt und in den Tab 'Optimierung' wechselt. In diesem Menü lassen sich die Start- und Endzustände auswählen, sowie wonach optimiert werden soll (Kosten oder Zeit) und wie oft optimiert werden soll (3 mal optimieren heißt, dass die besten 3 Wege angezeigt werden, diese Wege sind unter den Einstellungen aufgelistet). Auch in diesem Menü kann man die Optimierung starten, indem man auf 'Optimierung starten' (unten im Einstellungsmenü) klickt, sind Start- und Endzustände nicht ausgewählt, werden diese wieder automatisch bestimmt, die Optimierungsart (Zeit/Kosten) muss allerdings ausgewählt sein (alternativ dient der Button 'anwenden' zum übernehmen der Einstellungen und 'schließen' zum verwerfen der Einstellungen. In beiden Fällen schließt das Menü). Nach dem Betätigen des 'Optimierung starten' Buttons (im Einstellungsmenü) wird dieser ersetzt durch die besten Wege im Graphen (begrenzt durch Optimierungsanzahl). Durch klick auf einen dieser, erhält man einen Einblick in die einzelnen Zustände und Bearbeitungsschritte des Weges, durch klick auf den Kreis links vom angezeigten Weg, wird dieser anstatt dem besten Weg orange markiert. Sobald etwas am Graph geändert wird, erscheinen die 'Graph optimieren' bzw. 'Optimierung starten' Buttons wieder. Zuvor lässt sich der Graph durch Klicken auf das Wiederholen-Icon (neben 'Gesamtzeit' oder 'Gesamtkosten' im Graphenfenster) erneut optimieren.

- **Einstellungen**

Mit einem Linksklick auf 'Einstellungen' am oberen Rand öffnet sich das Einstellungsfenster. Im Tab 'Graph' lassen sich Einstellungen bezüglich der Darstellung des Graphs vornehmen, wie zum Beispiel die Einheiten oder die angezeigten Details der Verknüpfungen (Bearbeitungsschritte). Die Tabs 'Benutzer' und 'Hilfe' sind noch nicht funktional.

- **Offline Speicher**

Über den Button 'Export' am oberen Rand lässt sich der Graph in den Formaten .json, .png, .svg und .jpg exportieren und lokal speichern. Über den 'Import' Button lässt sich ein in .json exportierter Graph wieder im Programm öffnen.

- **Online Speicher**

Über den Button 'Datenbank' am oberen Rand öffnet sich das Datenbank-Fenster. Hier können erstellte Graphen online gespeichert bzw. wieder geladen werden. Alle hier gespeicherten Graphen werden mit dem genutzten Benutzernamen gespeichert und können nur von diesem Benutzer wieder geladen werden.

VIII.2 Installationsanleitung

Autor: xxx

XXX

VIII.3 Software-Lizenz

Autor: xxx

XXX

IX. PROJEKTABSCHLUSS

IX.1 Protokoll der Abnahme und Inbetriebnahme beim Kunden

Autor: xxx
XXX

IX.2 Präsentation auf der Messe

Autor: xxx
Poster, Bericht

IX.3 Abschließende Einschätzung durch Product-Owner

Autor: xxx
XXX

IX.4 Abschließende Einschätzung durch Software-Architekt

Autor: xxx
XXX

IX.5 Abschließende Einschätzung durch Team-Manager

Autor: xxx
XXX