

Master-Thesis

zur Erlangung des akademischen Grades

Master of Science (M.Sc.)

an der Hochschule für Technik und Wirtschaft des Saarlandes

im Studiengang Praktische Informatik

der Fakultät für Ingenieurwissenschaften

Konzeption und Umsetzung einer interaktiven Anwendung zur Darstellung von Sensordaten unter besonderer Berücksichtigung der User Experience

vorgelegt von

Johann Chopin

betreut und begutachtet von

Prof. Dr. Markus Esch

Berlin, Tag. Monat Jahr

Selbständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit (bei einer Gruppenarbeit: den entsprechend gekennzeichneten Anteil der Arbeit) selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ich erkläre hiermit weiterhin, dass die vorgelegte Arbeit zuvor weder von mir noch von einer anderen Person an dieser oder einer anderen Hochschule eingereicht wurde.

Darüber hinaus ist mir bekannt, dass die Unrichtigkeit dieser Erklärung eine Benotung der Arbeit mit der Note „nicht ausreichend“ zur Folge hat und einen Ausschluss von der Erbringung weiterer Prüfungsleistungen zur Folge haben kann.

Berlin, Tag. Monat Jahr

Johann Chopin

Zusammenfassung

Kurze Zusammenfassung des Inhaltes in deutscher Sprache, der Umfang beträgt zwischen einer halben und einer ganzen DIN A4-Seite.

Orientieren Sie sich bei der Aufteilung bzw. dem Inhalt Ihrer Zusammenfassung an Kent Becks Artikel: <http://plg.uwaterloo.ca/~migod/research/beckOOPSLA.html>.

*We have seen that computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity, and especially
because it produces objects of beauty.*

— Test [1] and Test [5]

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Auffertigung dieser Bachelorarbeit unterstützt und motiviert haben.

Zuerst gebührt mein Dank Herr Prof. Markus Esch die meine Masterarbeit betreut und begutachtet hat. Für die hilfreichen Anregungen und die konstruktive Kritik bei der Erstellung dieser Arbeit möchte ich mich herzlich bedanken.

Ebenfalls möchte ich mich bei meinen Kollegen und Kolleginnen der Firma Dryad bedanken (insbesondere mein Kollege Dinesh Priyashantha und mein Vorgesetzter Cherian Mathew), die mir während der Schreibphase meiner Masterarbeit wertvollen Input gegeben haben und mir Einblicke in die Unternehmenskultur gewährt haben.

Schließlich wäre es nachlässig, meine Familie nicht zu erwähnen, insbesondere meine Eltern. Ihr Glaube an mich hat mich während dieses Prozesses bei Laune und Motivation gehalten.

Chopin Johann

Inhaltsverzeichnis

1 Einleitung	1
1.1 Motivation	1
1.2 Aufgabenbeschreibung und Ziele	1
1.3 Aufbau der Thesis	1
2 Grundlagen	3
2.1 Silvanet-Technologie	3
2.2 Anwendungsentwicklung mit Webtechnologien	4
2.2.1 Single Page Application (SPA)	4
2.2.2 TypeScript	5
2.2.3 Webpack	5
2.2.4 Angular	6
2.2.5 Npm	7
2.2.6 SASS	7
2.2.7 Angular Routing	7
2.3 Tools und Arbeitsabläufe	8
2.3.1 Scrum	8
2.3.2 Tools	9
2.3.3 Developer Experience	10
2.4 Ergonomie und User Experience	11
2.4.1 Ergonomie	12
2.4.2 User Experience	12
3 Analyse	13
3.1 Zielsetzung	13
3.2 Ergonomische Inspektion	13
3.3 Definition von User Stories	13
4 Konzeption	15
4.1 Verwenden eines Dashboards für schnelle Entscheidungsfindung	15
4.2 Verhalten einer interaktiven Karte beim Zoomen und Auszoomen	15
4.3 Präsentation von Daten in Stresszeiten	15
5 Implementierung	17
5.1 Entwicklung eines Benutzers Fragebogens	17
5.2 Geschichtete Darstellung auf interaktiver Karte	17
6 Evaluation	19
7 Diskussion	21
8 Zusammenfassung und Ausblick	23
8.1 Zusammenfassung	23
8.2 Ausblick	23

Literatur	25
Abbildungsverzeichnis	27
Tabellenverzeichnis	27
Listings	27
Abkürzungsverzeichnis	29

1 Einleitung

1.1 Motivation

Jedes Jahr und immer häufiger brechen auf der ganzen Welt immer intensivere und verheerendere Waldbrände aus. Nur in Europa brannten 2021 mehr als 1.113.464 ha Wald in mehr als 39 Ländern [3]. Um das Risiko von Waldbränden zu verringern, hat das junge Start-up-Unternehmen Dryad Sensoren entwickelt, die eine ultraschnelle Erkennung von Waldbränden ermöglichen.

Diese Sensoren, die über mehrere Aktionsstandorte verteilt sind, erfassen verschiedene Arten von Daten wie Temperatur, Luftfeuchtigkeit und das Vorhandensein von Gasen, die bei einer Verbrennung entstehen. Ein Webinterface ist verfügbar, um die verschiedenen Daten, die von diesen Sensoren gesendet werden, zu sehen. Da die Kunden jedoch nicht alle Computerspezialisten sind, muss die Benutzeroberfläche so intuitiv wie möglich sein.

Derzeit kommt das Produkt mit einer Anleitung, die erklärt, wie man die Webanwendung nutzt. Dies deutet auf ein großes Problem mit der Nutzererfahrung hin. Meine Aufgabe in dieser Arbeit ist es daher, die Schwachstellen der aktuellen Schnittstelle zu identifizieren und Verbesserungen vorzuschlagen und zu entwickeln.

1.2 Aufgabenbeschreibung und Ziele

In meiner Thesis bei Dryad geht es darum, die entscheidenden Bereiche der Webanwendung, die die verschiedenen Sensoren verwaltet, zu erkennen, die als schlechte Benutzererfahrung angesehen werden. Die Schwerpunkte der Arbeit lassen sich wie folgt beschreiben:

- Identifizierung von ergonomischen Mängeln basierend auf einer Auditierung der gesamten Schnittstelle.
- Entwicklung und Durchführung von Benutzertests, die es ermöglichen, User-Stories so zu definieren, dass sie nach Prioritäten geordnet werden können.
- Forschung und Entwicklung zu den verschiedenen interaktiven Karten der Anwendung, die es dem Nutzer erleichtern, seine Standorte und Sensoren sowie deren Status zu entdecken.
- Forschung zur Verbesserung der Datenpräsentation in Stresssituationen

1.3 Aufbau der Thesis

In Kapitel 2 werden zunächst die grundlegenden Konzepte vorgestellt, die für das Verständnis der These hilfreich sind. In Kapitel 3 werden dann die Anforderungen vorgestellt, die für diese Thesis definiert wurden.

[TODO]

2 Grundlagen

2.1 Silvanet-Technologie

Das Startup namens Dryad¹ mit Sitz in Berlin hat sich zum Ziel gesetzt, die Zahl der Waldbrände zu verringern, indem es sie so schnell wie möglich entdeckt und die Behörden warnt. Ein Team für die Entwicklung von Hardware und eingebetteten Systemen beschäftigt sich mit der Entwicklung kleiner solarbetriebener Sensoren, die Veränderungen in der Zusammensetzung der Umgebungsluft erkennen können. Wenn eine Veränderung festgestellt wird, wird ein Scan initiiert und eine eingebaute künstliche Intelligenz bestimmt, ob es sich um die Verbrennung handelt. Der Vorteil der künstlichen Intelligenz besteht darin, dass sie mehrere Variationen als nur die einfache Verbrennung erkennen kann. So kann man feststellen, ob es sich nur um einen Diesel-LKW handelt, der in der Nähe vorbeifährt, oder ob es sich um ein Feuer aus einem bestimmten Holz handelt, etc.

Diese Sensoren werden mithilfe der LoRaWAN-Technologie miteinander verbunden, die es ermöglicht, ein großes Netzwerk mit geringem Stromverbrauch aufzubauen. Die Daten der Sensoren werden über Gateways an den Backend-Dienst von dryad gesendet, der sie sammelt und verarbeitet. Nach der Verarbeitung können die verschiedenen Informationen der zu Standorten zusammengefassten Sensoren abgerufen und in der Webanwendung von dryad angezeigt werden. Mit dieser Anwendung kann sich ein Benutzer anmelden, die Installation von Sensoren an einem Standort planen, die verschiedenen gesammelten Daten in Echtzeit verfolgen und die Art einer Waldbrandwarnung im Detail abrufen.

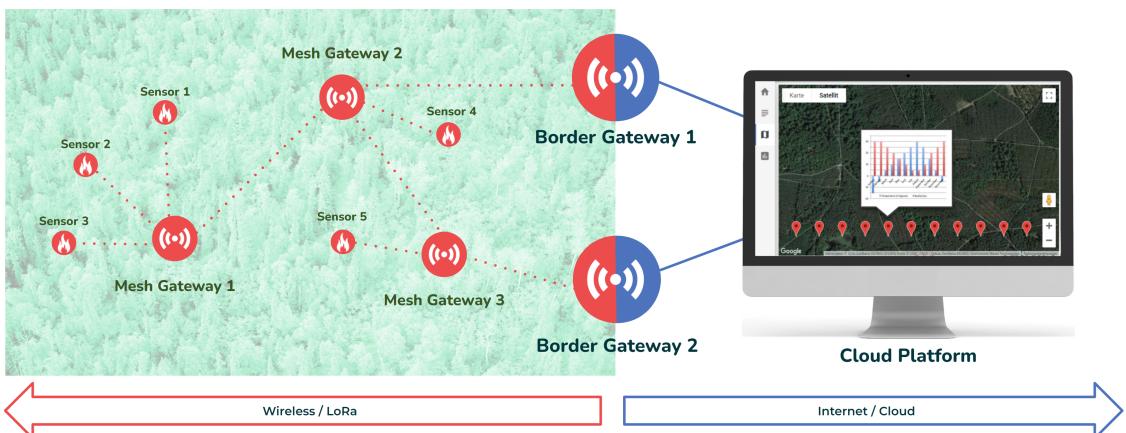


Abbildung 2.1: Schematische Darstellung des LoRaWAN-Netzwerks, das die verschiedenen Sensoren mit dem Cloud-System verbindet.

¹<https://www.dryad.net/>



Abbildung 2.2: ein luftschnüffelnder Sensor, der ein Silvanet-Netzwerk bildet

2.2 Anwendungsentwicklung mit Webtechnologien

Die Sensorverwaltungsanwendung von dryad wird mithilfe von Webtechnologien (html, css, javascript) und zahlreichen Tools entwickelt, um eine Webanwendung zu erstellen, die auf mehreren Plattformen (Browser, ios, Android) läuft und dabei nur einen einzigen Quellcode hat. Es wurde beschlossen, dass die erste Version der Anwendung eine Single Page Application (SPA) sein sollte. Derzeit gibt es viele Tools, die das Erstellen einer SPA erleichtern. Innerhalb von Cloud-Team von Dryad wurde beschlossen, die folgenden zu verwenden:

2.2.1 Single Page Application (SPA)

Eine Single Page Application ist ein Website-Design-Ansatz, bei dem der Inhalt jeder neuen Seite nicht durch das Laden neuer HTML-Seiten, sondern dynamisch durch die Fähigkeit von JavaScript, die Document Object Model (DOM)-Elemente auf der bestehenden Seite selbst zu manipulieren, generiert wird.

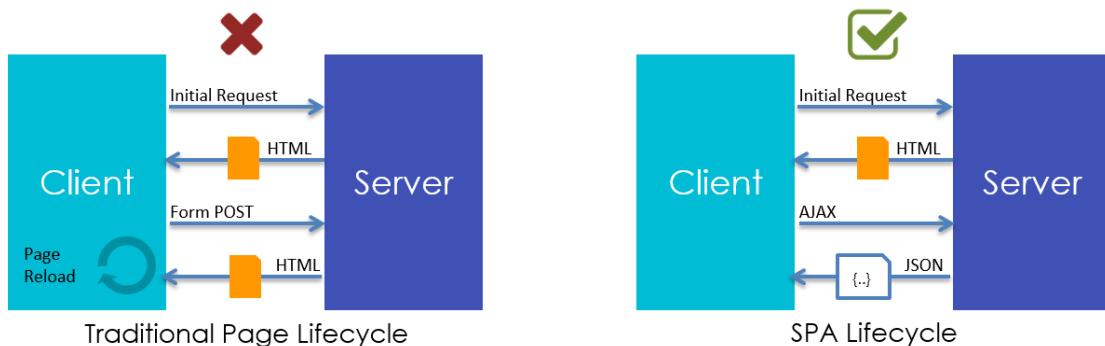


Abbildung 2.3: <https://www.tothenew.com/blog/optimization-of-angularjs-single-page-applications-for-web-crawlers/>

Pros:

- Die traditionellen HTML-Seiten nehmen beim Hin- und Herbewegen zum Server viel Zeit in Anspruch. Dies erforderte das Nachladen der großen Menge ähnlicher Daten, während das neue SPA die Notwendigkeit der Kommunikation mit HTML-Tags zum Server mildert. SPA lädt nur einmal mit den HTML-Tags, während die nächsten Anfragen auf das teilweise Laden der Seite mittels AJAX und JSON erfolgen. Dies führt zu einer Einsparung von Bandbreite.

- Die SPA-Seite wird schneller geladen und benötigt aufgrund der geringeren Datentransaktion weniger Bandbreite. Das Seitendesign und UX werden besser und funktionieren erstaunlich gut mit den langsamten Verbindungen.

Cons:

- Die traditionellen HTML-Seiten sind gut für den SEO, während die SPA schwer von den Benutzern gecrawlt werden kann, weil die Webcrawlers nicht wissen, wie sie mit dem Javascript umgehen sollen. Die Lösung besteht darin, die roboterspezifische HTML-Seite zu kodieren, was wiederum zu Wartungsproblemen führen kann.
- Manchmal braucht die SPA viel Zeit, um das Bündel aus CSS und Javascript zu laden, was das erste Laden der Seite verzögert.

2.2.2 TypeScript

Die Entwicklung einer Webanwendung kommt heutzutage ohne Javascript kaum noch aus. Die Hegemonie dieser Sprache wird nicht mehr angezweifelt, und ihre Gemeinschaft ist eine der wichtigsten in der Welt der Webentwicklung, was sicherlich auf ihre inhärente Flexibilität zurückzuführen ist. Es hat jedoch einige Einschränkungen, wenn es um die Entwicklung komplexer Anwendungen geht, und aus diesem Grund kommt TypeScript ins Spiel.[6]

TypeScript² ist eine Programmiersprache, die von Microsoft im Jahr 2012 entwickelt wurde. Sein Hauptanliegen ist es, die Produktivität der komplexen Anwendungsentwicklung zu verbessern.

Es handelt sich um eine Open-Source-Sprache, die als höhere Schicht von Javascript entwickelt wurde. Jeder in Javascript gültige Code ist auch in TypeScript gültig. Die Sprache führt jedoch optionale Funktionen wie Typisierung oder objektorientierte Programmierung ein. Um diese Funktionen nutzen zu können, ist keine Bibliothek erforderlich, aber nur das Tool für die TypeScript-Kompilierung. Somit wird der ausgeführte Code ein Javascript-Äquivalent des kompilierten TypeScript-Codes sein.

Dieser Sicherheitstyp hat ihn bei Javascript-Entwicklern sehr beliebt gemacht, wie die Stackoverflow Survey 2022³ beweist:

2.2.3 Webpack

Webpack⁴ ist ein Modul-Bündler. Sein Hauptzweck besteht darin, JavaScript-Dateien für die Verwendung in einem Browser zu bündeln, aber es ist auch in der Lage, so gut wie jede Ressource oder jedes Asset zu transformieren, zu bündeln oder zu paketieren⁵.

Es gibt viele verschiedene Bündler wie Parceljs⁶, Rollupjs⁷ oder Browserify⁸. Das von uns verwendete Web-Framework (das im nächsten Abschnitt vorgestellt wird) kommt jedoch fertig mit einer webpack-Konfiguration. So wird webpack neben seiner Reife, seinen zahlreichen Plug-ins und seiner umfangreichen Konfiguration auch für den Aufbau unserer Javascript-Anwendung verwendet.

²<https://www.typescriptlang.org/>

³<https://survey.stackoverflow.co/2022/#most-loved-dreaded-and-wanted-language-want>

⁴<https://github.com/webpack/webpack>

⁵Offizielle Webpack-Dokumentation: <https://github.com/webpack/webpack#webpack>

⁶<https://parceljs.org/>

⁷<https://github.com/rollup/rollup>

⁸<https://github.com/browserify/browserify>

2 Grundlagen

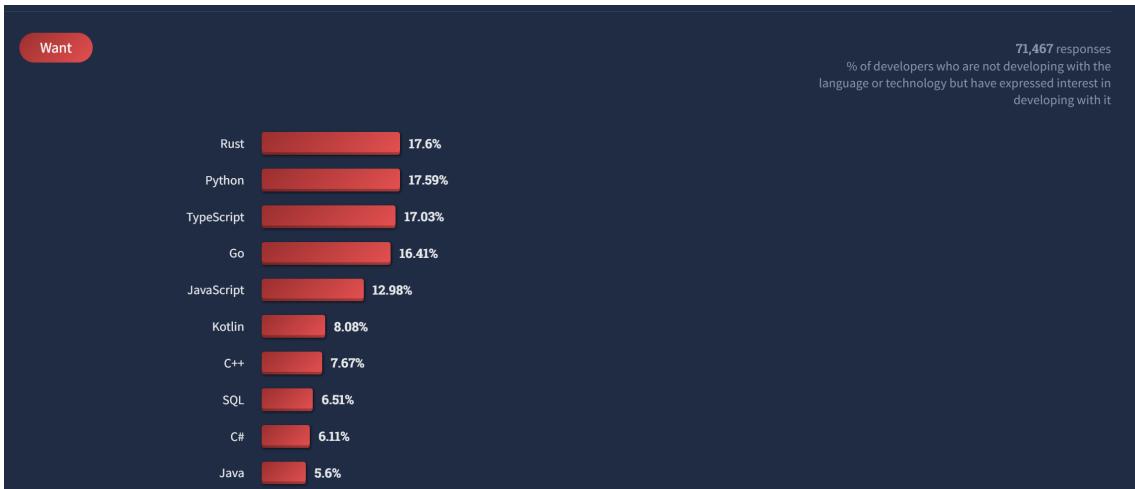


Abbildung 2.4: Liste der beliebtesten Programmier-, Skript- und Auszeichnungssprachen im Jahr 2022

2.2.4 Angular

Mit Javascript können wir ein HTML DOM leicht manipulieren. Das Hinzufügen, Ändern oder Löschen eines Node zur Laufzeit, ohne die Seite neu zu laden, bietet dem Benutzer eine sehr dynamische und leichte Schnittstelle, die dem ähnelt, was er in einer nativen Anwendung sehen kann.

Leider kann die Entwicklung einer Webanwendung allein mithilfe von Javascript sehr viel Zeit in Anspruch nehmen und schwer skalierbar werden. Aus diesem Grund haben sehr marktführende Unternehmen wie Google oder Facebook ihr Javascript-Framework entwickelt, um dynamische Schnittstellen einfach zu erstellen.

Ein Framework ist eine semi-vollständige Applikation. Es stellt für Applikationen eine wiederverwendbare, gemeinsame Struktur zur Verfügung. Die Entwickler bauen das Framework in ihre eigene Applikation ein und erweitern es derart, dass es ihren spezifischen Anforderungen entspricht. Frameworks unterscheiden sich von Toolkits dahingehend, dass sie eine kohärente Struktur zur Verfügung stellen, anstatt einer einfachen Menge von Hilfsklassen[4].

Einige der beliebtesten sind React⁹ von Facebook, Angular¹⁰ von Google und Vue¹¹.



Abbildung 2.5: Stackoverflow Survey 2022: Beliebteste Web-Frameworks

⁹<https://github.com/facebook/react/>

¹⁰<https://github.com/angular/angular>

¹¹<https://github.com/vuejs/vue>

Angular wurde aufgrund seines umfassenden Designs, seiner großen Nutzergemeinde und der Tatsache, dass das Team bereits über fundierte Kenntnisse in diesem Tool verfügte, als Framework ausgewählt.

2.2.5 Npm

Es ist derzeit undenkbar, eine Webapplikation von Grund auf neu zu entwickeln. Es gibt eine Vielzahl von Open-Source-Bibliotheken, die von verschiedenen Entwicklern entwickelt wurden und mit denen unsere Anwendung schnell und einfach erstellen werden kann. Deshalb ist es notwendig, ein Tool zu benutzen, um die verschiedenen Bibliotheken, die verwendet werden, zu organisieren.

Node Package Manager (NPM) ist das Zentrum der gemeinsamen Nutzung von JavaScript-Code und mit mehr als einer Million Paketen und 11 Millionen Entwickler weltweit die größte Software-Registry der Welt¹². Aufgrund dieses guten Renommes beschloss das Team, NPM zur Verwaltung der verschiedenen Bibliotheken einzusetzen.

2.2.6 SASS

Angular erlaubt es, das DOM einer Webanwendung leicht zu manipulieren, bleibt nur noch, sie zu stylen. Es wird dafür der Stylesheet-Kernsprachen des World Wide Webs verwenden, Cascading Style Sheets (CSS).

Leider kann CSS schnell überflüssig und schwierig zu handhaben werden, wenn es darum geht, mehrere Dutzend Elemente gleichzeitig zu bearbeiten. Um dieses Problem zu überwinden, wurde der CSS-Präprozessor Syntactically Awesome StyleSheets (SASS) geschaffen, der die Verwendung von Variablen, Schleifen und Funktionen erlaubt, die in CSS umgesetzt werden.

Die folgende Abbildung zeigt die gleiche Style-Deklaration mit CSS und SASS:

Tabelle 2.1: Beispiel für die Deklaration von Styles in css und sass

CSS	SASS
<pre>ul { background-color: red; } ul.hide { opacity: 0; } ul.hide li { text-decoration: underline; }</pre>	<pre>ul { background-color: red; } &.hide { opacity: 0; li { text-decoration: underline; } }</pre>

Es wurde daher beschlossen, diesen css-Prozessor im Projekt zu verwenden. Außerdem bietet Angular eine bereits fertige Sass-Kompilation an.

2.2.7 Angular Routing

Der Antrag wird also von einer einzigen Seite generiert, was aber nicht bedeutet, dass ein Wechsel der URLs nicht möglich ist. Tatsächlich kann eine SPA dank Javascript auf

¹²npmjs offizielle Website: <https://www.npmjs.com/>

2 Grundlagen

verschiedenen URLs laufen. Angular bringt ein eigenes Routing-System namens Angular Routing¹³ mit, das es ermöglicht, bestimmte Komponenten auf der Grundlage der URL anzuzeigen. Das Projekt wird diese Lösung verwenden, um das Routing der Anwendung zu verwalten.

2.3 Tools und Arbeitsabläufe

Innerhalb des Cloud-Team wird jedes IT-Projekt agil nach der Scrum-Methode entwickelt. Scrum ist ein Framework für die Entwicklung komplexer Softwareprodukte. Sie wird von ihren Schöpfern als "ganzheitlicher, iterativer Rahmen definiert, der sich auf gemeinsame Ziele konzentriert, indem er produktiv und kreativ Produkte von höchstmöglichen Wert liefert"[7]. 13 Es basiert auf der Unterteilung eines Projekts in SZeit Boxen", sogenannte Sprints, die von ein paar Stunden bis zu einem Monat dauern können (in unserem Team dauert es eine Woche)

2.3.1 Scrum

Das Cloud-Team wendet mehrere Scrum-Methoden zur Verbesserung der Software-Entwicklung an:

2.3.1.1 Sprint Meeting

Zu Beginn jedes Sprints trifft sich das Cloud-Team, um die verschiedenen Aufgaben für den nächsten Sprint zu planen und diese in Tickets zu organisieren. Sie finden jeden Montag um 10 Uhr morgens statt und dauern durchschnittlich 30 Minuten.

2.3.1.2 Retro Meeting

Diese Treffen finden einmal pro Woche statt und ermöglichen jedem Teamentwickler, die guten und schlechten Punkte der vergangenen Woche zum Ausdruck zu bringen. Dies gibt Rückmeldung über die Stimmung im Team. Sie finden jeden Donnerstag um 13.00 Uhr statt und dauern durchschnittlich 45 Minuten.

2.3.1.3 Kanban board

Die Verfolgung unserer Sprint-Tickets erfolgt über eine Kanban-Tafel. Es handelt sich dabei um ein agiles Projektmanagement-Tool, das dazu dient, die Arbeit zu visualisieren, die laufende Arbeit zu begrenzen und die Effizienz zu maximieren. Unsere Tabelle besteht aus 6 Spalten:

- **Backlog:** Tickets geplant, aber nicht bearbeitet
- **Open:** Tickets, die im aktuellen Sprint erledigt werden müssen
- **On Hold:** Geplantes, aber für unbestimmte Zeit blockiertes Ticket
- **In Progress:** Die Tickets, die realisiert werden
- **Review:** Tickets, die derzeit einer Code Review unterzogen werden
- **Closed:** Tickets, die validiert und im Entwicklungszweig zusammengeführt wurden

¹³<https://angular.io/guide/routing-overview>

Jedes Ticket muss begleitet sein von:

- **Ein Story point:** es handelt sich um eine Zahl zwischen 1 und 8, die der geschätzten Anzahl der Stunden entspricht, die für die Erstellung des Tickets benötigt werden
 - **eine Priorität:** dem Ticket muss Vorrang eingeräumt werden (low, medium oder high)

2.3.2 Tools

Wir verwalten all diese Prozesse sowie die Verwaltung des Codes mit verschiedenen Tools:

2.3.2.1 Git

Git¹⁴ ist ein freies und quelloffenes verteiltes Versionskontrollsystem. Es handelt sich um ein einfaches und leistungsfähiges Tool, dessen Hauptaufgabe darin besteht, die Entwicklung der Inhalte einer Baumstruktur zu verwalten.

Git ist das Tool, aber um ein kollaboratives Projekt richtig zu verwalten, braucht das Team einen Arbeitsablauf. Das Team hat sich für Gitflow¹⁵ entschieden. Dieses Workflow definiert ein striktes branch-modell, das um eine Projektfreigabe herum entworfen wurde

Das innerhalb des Unternehmens verwendete Modell ist recht einfach. Es besteht aus einem "mainBranch, der die für die Code-Produktion fertige Version mit den verschiedenen Versionen enthält, den "developBranch, der dem Zweig entspricht, der sich auf alle "featureBranch bezieht:

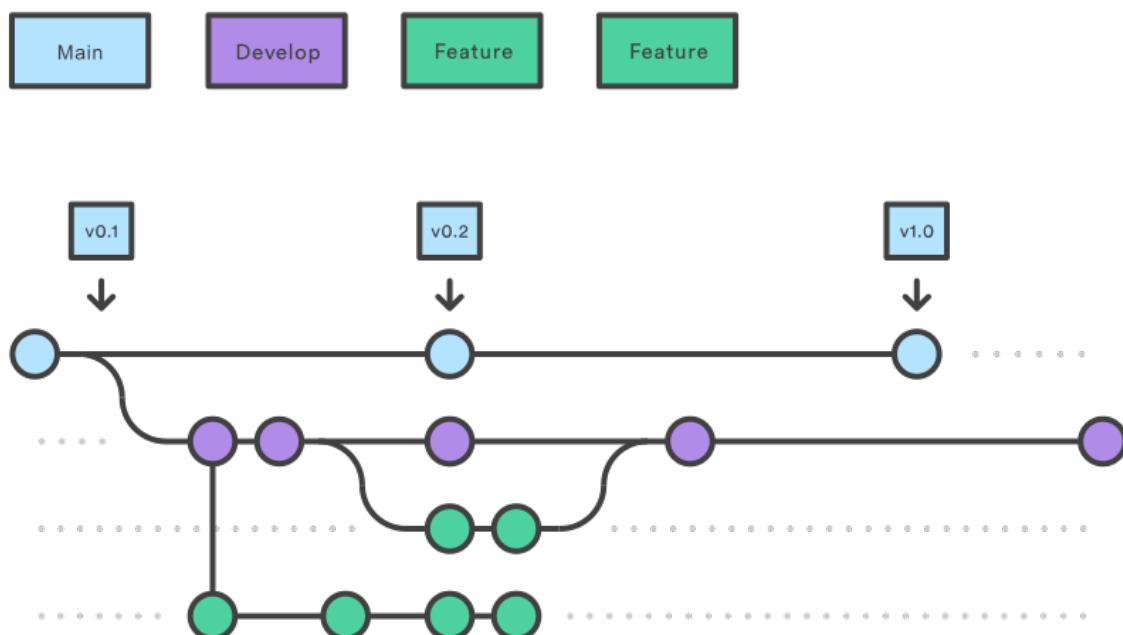


Abbildung 2.6: Schematische Darstellung eines Git-Baums, der dem Standard von gitflow entspricht

¹⁴<https://git-scm.com/>

¹⁵<https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

¹⁶Stackoverflows git-flow tag Beschreibung: <https://stackoverflow.com/questions/tagged/git-flow>

2 Grundlagen

Das Cloud-Team nahm eine Änderung am Schema vor und entfernte den branch develop, der angesichts der geringen Größe des Teams nicht relevant war.

2.3.2.2 Slack

Die Kommunikation innerhalb der Firma erfolgt mithilfe der Slack application¹⁷. Damit können wir einander leicht anrufen und unsere Sofortnachrichten auf verschiedene Kanäle organisieren.

2.3.2.3 Clickup

Clickup¹⁸ ermöglicht es uns, unseren agilen Prozess zu virtualisieren: Tickets, Kanban-Tafel, Sprint-Planung, Roadmap, ...

2.3.2.4 Google Workspace

Die Dryad-Teams verwenden Google Workspace-Tools, um die externe Kommunikation über Google Mail, die Finanzen über Google Sheets, die Speicherung verschiedener Dateien und Bilder mit Google Drive und die Bearbeitung von Textdateien mit Google Doc zu verwalten.

2.3.2.5 GitHub

Die Verwendung von GitHub ermöglicht es, die Git-Projekte des Unternehmens online zu hosten. Es ermöglicht uns die einfache Verwaltung unseres Quellcodes in Zusammenarbeit mit verschiedenen Features wie Code-Reviews, Veröffentlichung von npm-Paketen und kontinuierlicher Integration.

2.3.3 Developer Experience

Da der Tech Stack definiert ist, ist es möglich, das Projekt zu starten und etwas aufzubauen. Es kann jedoch Bedenken geben, wenn mehrere Entwickler gleichzeitig am gleichen Quellcode arbeiten (Konflikte, Code-Formatierung, Sicherstellung der Gültigkeit des Codes, ...). Deshalb ist es wichtig, bestimmte Regeln und Tools einzuführen, die es dem Entwickler ermöglichen, sich auf die eigentliche Aufgabe zu konzentrieren. Für die Schaffung dieses Projekts war es wesentlich, die folgenden Tools/Prozesse zu implementieren, um sich in einer einfachen und homogenen Entwicklungsumgebung leicht weiterentwickeln zu können.

2.3.3.1 Linter

Ein Linter ist ein AnalyseTool für den statischen Code, das zur Kennzeichnung von Programmierfehlern, Bugs, stilistischen Fehlern und verdächtigen Konstrukten verwendet wird. Es wurde beschlossen, den eslint¹⁹ Javascript-Linter wegen seiner Popularität und seiner verschiedenen, leicht konfigurierbaren Plugins zu verwenden (Angular²⁰, TypeScript²¹, ...).

¹⁷<https://slack.com/>

¹⁸<https://clickup.com/>

¹⁹<https://github.com/eslint/eslint>

²⁰<https://github.com/angular-eslint/angular-eslint>

²¹<https://github.com/typescript-eslint/typescript-eslint>

Eslint erlaubt es, den von jedem Entwickler geschriebenen Code zu homogenisieren. So ist die Einrückung für alle gleich, ebenso wie Zeilenumbrüche, Leerzeichen, variable Notation, usw...

```
> NX Ran target lint for project silvanet-web (7s)
  With additional flags:
    --fix=true

  ✘ 1/1 failed
  ✓ 0/1 succeeded [0 read from cache]

marker.ts
  45:7 error 'justAnUnusedVariable' is assigned a value but never used @typescript-eslint/no-unused-vars
✖ 1 problem (1 error, 0 warnings)
```

Abbildung 2.7: Beispiel für die Linter-Ausführung, die einen Fehler im Terminal zurückgibt

2.3.3.2 TypeScript

TypeScript ist ein Tool, das entwickelt wurde, um Entwicklern die einfache Realisierung von JavaScript-Projekten zu erleichtern. Da es aber sehr anpassungsfähig ist, muss jedoch bekannt sein, was von diesem Tool erwartet wird und wie es verwendet werden soll.

Es wurde daher beschlossen, TypeScript mit einer hohen Anforderung zu konfigurieren. Daher muss alles typisiert werden (Variablen, Funktionen, Parameter), damit jeder Entwickler leichter verstehen kann, was an jeder Stelle im Code geschieht. Darüber hinaus ist der "beliebige" Typ verboten, da er die Verwendung von TypeScript überflüssig macht.

Dies mag dem DX-Prinzip zuwiderlaufen, da es anfangs viele Einschränkungen mit sich bringt, aber nach einer kurzen Anpassungszeit kommen die Vorteile eines stark typisierten Codes schnell zum Tragen und verbessern die Erfahrung des Entwicklers erheblich.

2.3.3.3 Drone CI

In jedem Beruf ist es sehr ärgerlich, immer wieder die gleichen Aufgaben auszuführen. In der Entwicklungswelt müssen wir zum Beispiel sicherstellen, dass der zur Produktion geschickte Code dem Linters-Standard entspricht, sich problemlos kompilieren lässt und vorher alle Tests besteht. Ständig darauf achten zu müssen, sie nicht zu vergessen, belastet die Entwickler unnötig mental.

Aus diesem Grund verwendet das Unternehmen ein kontinuierliches Integrations-Tool, Drone CI²². Es ist sehr kompatibel mit GitHub und ermöglicht, bestimmte Aufgaben auszuführen, wenn Code auf GitHub gepusht wird.

Seine große Stärke ist die Einfachheit der Konfiguration. Es muss nämlich nach der Verknüpfung von travis mit dem GitHub-Projekt nur noch eine einfache `.drone.yml`-Datei an der Root des Projekts erstellt und bearbeitet werden.

2.4 Ergonomie und User Experience

Es gibt heute viele Möglichkeiten, eine Benutzeroberfläche zu präsentieren. Viele Schnittstellen vermitteln jedoch den Eindruck, dass sie schwer zu erforschen und zu nutzen sind, und es ist schwierig, das ursprüngliche Ziel zu erreichen. So mussten Konzepte erstellt werden, um einerseits eine Schnittstelle zu entwerfen und andererseits eine Schnittstelle,

²²<https://www.drone.io/>

2 Grundlagen

die von einem Menschen genutzt wird, so zu bewerten, dass sie effektiv, effizient und mit denen die Nutzer zufrieden sein werden. So entstanden zwei Designkonzepte, die seither allgemein verwendet werden: Ergonomie und User Experience (UX) von Schnittstellen.

2.4.1 Ergonomie

Ergonomie kann definiert werden als "Ausmaß, in dem ein Produkt von bestimmten Benutzern verwendet werden kann, um bestimmte Ziele mit Effektivität, Effizienz und Zufriedenheit in einem bestimmten Nutzungskontext zu erreichen"[1]. Da es sich um eine recht umfassende und allgemeine Definition handelt, kann man sie in zwei Konzepte unterteilen.

Eine der Konzeptionen ist, dass sich die Ergonomie auf Maßnahmen konzentrieren sollte, die mit der Erfüllung der übergeordneten Ziele der Aufgabe zusammenhängen. Dies beinhaltet Techniken zur summativen Bewertung oder messbasierten Bewertung, sowohl objektive als auch subjektive Schnittstellen.

Die andere Konzeption ist, dass sich die Praktiker auf die Erkennung und Beseitigung von Problemen der Benutzerfreundlichkeit konzentrieren sollten.

2.4.2 User Experience

Während die Ergonomie also technisch und wissenschaftlich orientiert ist, um einen Benutzer bei der Ausführung einer Aufgabe zu bewerten, ist die UX emotionaler.

So sind instrumentelle Attribute wie Effizienz und Effektivität immer noch wichtig, aber hauptsächlich in dem Maße, in dem sie emotionale Ergebnisse wie Zufriedenheit, Vertrauen und wahrgenommene Schönheit beeinflussen, mit daraus resultierenden Auswirkungen auf das Ergebnisverhalten wie z. B. wiederholte Käufe und Weiterempfehlung an andere Nutzer.

UX nach Geoffrey Crofte [2] ist daher ganzheitlich und zeitlich orientiert, d. h. vor, während und nach der Nutzung der Schnittstelle. Daher konzentriert sich UX auf die Faktoren, die auf die Nutzung einer Schnittstelle folgen:

- das soziale Image der erwarteten Erfahrung
- die Erinnerung an vergangene Erlebnisse
- das soziale Teilen von Erfahrungen
- die Kontinuität des Service
- die Erinnerung an die durchgeföhrte Erfahrung

3 Analyse

3.1 Zielsetzung

3.2 Ergonomische Inspektion

3.3 Definition von User Stories

4 Konzeption

- 4.1 Verwenden eines Dashboards für schnelle Entscheidungsfindung**
- 4.2 Verhalten einer interaktiven Karte beim Zoomen und Auszoomen**
- 4.3 Präsentation von Daten in Stresszeiten**

5 Implementierung

5.1 Entwicklung eines Benutzers Fragebogens

5.2 Geschichtete Darstellung auf interaktiver Karte

6 Evaluation

7 Diskussion

8 Zusammenfassung und Ausblick

8.1 Zusammenfassung

8.2 Ausblick

Literatur

- [1] J. M. C. Bastien und D. L. Scapin. „Ergonomic Criteria for the Evaluation of Human-Computer Interfaces“. In: (1993).
- [2] Geoffrey Crofte. „Ergonomie des interfaces et Expérience Utilisateur“. In: (2017).
- [3] San-Miguel-Ayanz J, Durrant T, Boca R, Maianti P, Liberta' G, Artes Vivancos T, Jacome Felix Oom D, Branco A, De Rigo D, Ferrari D, Pfeiffer H, Grecchi R und Nuijten D. „Advance report on wildfires in Europe, Middle East and North Africa 2021“. In: KJ-NA-31028-EN-N (online) (2022). ISSN: 1831-9424 (online). DOI: 10.2760/039729 (online).
- [4] Ralph E. Johnson und Brian Foote. „Designing Reusable Classes“. In: *Journal of Object-Oriented Programming* (1988).
- [5] J. R. Lewis und J. Sauro. „Usability and User Experience: Design and Evaluation“. In: (2021).
- [6] Christopher Nance. „TypeScript Essentials“. In: *Packt Publishing Ltd* (2014), S. 27–31.
- [7] Ken Schwaber und Jeff Sutherland. „The Scrum Guide“. In: (2010).

Abbildungsverzeichnis

2.1	Schematische Darstellung des LoRaWAN-Netzwerks, das die verschiedenen Sensoren mit dem Cloud-System verbindet.	3
2.2	ein luftschnüffelnder Sensor, der ein Silvanet-Netzwerk bildet	4
2.3	https://www.tothenew.com/blog/optimization-of-angularjs-single-page-applications-for-web-crawlers/	4
2.4	Liste der beliebtesten Programmier-, Skript- und Auszeichnungssprachen im Jahr 2022	6
2.5	Stackoverflow Survey 2022: Beliebteste Web-Frameworks	6
2.6	Schematische Darstellung eines Git-Baums, der dem Standard von gitflow entspricht	9
2.7	Beispiel für die Linter-Ausführung, die einen Fehler im Terminal zurückgibt	11

Tabellenverzeichnis

2.1	Beispiel für die Deklaration von Styles in css und sass	7
-----	---	---

Listings

Abkürzungsverzeichnis

CSS Cascading Style Sheets

DOM Document Object Model

NPM Node Package Manager

SASS Syntactically Awesome StyleSheets

UX User Experience

Anhang

Kolophon

Dieses Dokument wurde mit der L^AT_EX-Vorlage für Abschlussarbeiten an der htw saar im Bereich Informatik/Mechatronik-Sensortechnik erstellt (Version 1, Juni 2022). Die Vorlage wurde von Yves Hary und André Miede entwickelt (mit freundlicher Unterstützung von Thomas Kretschmer, Helmut G. Folz und Martina Lehser). Daten: (F)10.95 – (B)426.79135pt – (H)688.5567pt