

Beschte Onboarding

Julian Thiele

18. November 2023

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einleitung | 3 |
| 1.1 | Ausbildungsbetrieb | 3 |
| 1.2 | Projektumfeld | 3 |
| 1.3 | Projektabgrenzung | 3 |
| 1.4 | Abweichungen vom Projektantrag | 3 |
| 2 | Projektplanung | 3 |
| 2.1 | Identifizierung der Arbeitspakete | 4 |
| 2.2 | Zeitplan | 4 |
| 2.3 | Ressourcenplanung | 4 |
| 2.4 | Kostenplanung | 5 |
| 2.5 | Nutzwertanalyse | 5 |
| 3 | Analysephase | 6 |
| 3.1 | Anforderungsgespräch | 6 |
| 3.2 | IST-Zustand Analyse | 7 |
| 3.3 | SOLL-Konzept | 7 |
| 4 | Designphase | 8 |
| 4.1 | Benutzeroberfläche | 8 |
| 4.2 | Datenbank | 8 |
| 5 | Implementierungsphase | 9 |
| 5.1 | Frontend | 9 |
| 5.2 | Backend | 10 |
| 6 | Testphase | 11 |
| 7 | Fazit | 11 |
| 7.1 | Soll/Ist-Vergleich | 11 |
| 7.2 | Lessons learned | 12 |

1 Einleitung

1.1 Ausbildungsbetrieb

Die BREDEX GmbH wurde im Jahr 1987 in Braunschweig gegründet. Den Schwerpunkt bildet die individuelle Softwareentwicklung, die BREDEX führt jedoch auch Beratungen zu Datenschutz, Datensicherheit, Qualitätssicherung von Software und Schulungen durch. Die BREDEX GmbH besitzt, mit seiner Tochterfirma BREDEX HUNGARY KFT. zusammen, ca. 200 Mitarbeiter, mit ungefähr 10 Auszubildenden und dualen Studenten.

1.2 Projektumfeld

Das Projekt wird unter der Aufsicht von Herr Christian Rucinski in den Geschäftsräumen der BREDEX GmbH entwickelt. Bei der Anwendung handelt es sich um ein BREDEX internes Tool. Sie soll für den Onboardingprozess neuer Mitarbeiter eingesetzt werden und den sogenannten BREDEX Buddy, einen Ansprechpartner neuer Mitarbeiter, bei seiner Arbeit unterstützen.

Das Frontend wird mit dem Framework Vue, bei welchem die Scriptsprache TypeScript Anwendung findet, geschrieben. Als Backend wird Firebase als "Backend As a Service" genutzt, in welche die NoSQL-Datenbank Firestore integriert ist.

1.3 Projektabgrenzung

Durch die Fachinformatikerausbildungsverordnung wird das Projekt zeitlich auf 80 Stunden begrenzt.

Die Anwendung ist ein alleinstehendes Tool und wird daher in kein Projekt eingegliedert. Daher gibt es keine weiteren Beschränkungen, auf die geachtet werden muss.

1.4 Abweichungen vom Projektantrag

Es gibt keine Abweichungen vom Projektantrag.

2 Projektplanung

Bei der Projektplanung werden die benötigten Arbeitspakete identifiziert und geplant. Auch benötigte Kosten und Ressourcen werden abgeschätzt.

2.1 Identifizierung der Arbeitspakete

Zu Beginn findet ein Anforderungsgespräch statt, bei dem die Projektziele definiert werden, damit es keine Differenzen im Verständnis dieser gibt. Außerdem ist hierdurch definiert, welche Anforderungen erfüllt werden müssen.

Im Anschluss kommt die Analysephase, bei der der Soll-Zustand niedergeschrieben wird.

Das dritte Paket bildet die Designphase, hierbei werden Benutzeroberfläche und Datenmodell geplant.

Daraufhin folgt die Entwicklung der Anwendung, bei der die Ergebnisse aus den vorherigen Arbeitspaketen umgesetzt werden. Um mögliche Fehler während der Implementierung frühzeitig zu erkennen und beheben zu können, finden Tests statt.

Im letzten Arbeitsschritt wird die Dokumentation des Projektes erstellt.

2.2 Zeitplan

Das Projekt wird im Zeitraum vom 22.09.2023 bis zum 22.11.2023 bearbeitet. Die im vorherigen Abschnitt erwähnten Arbeitspakete wurden in folgenden Zeitplan (Tabelle 1) eingeteilt.

| Arbeitspaket | Dauer (in Stunden) |
|----------------------|-----------------------|
| Anforderungsgespräch | 3 |
| Analysephase | 6 |
| Planungsphase | 6 |
| Designphase | 6 |
| Umsetzung | 40 |
| Test und Abnahme | 7 |
| Dokumentation | 12 |
| Gesamt | 80 |

Tabelle 1: Zeitplan

2.3 Ressourcenplanung

Um die Kosten so gering wie möglich zu halten, wurde nach Möglichkeit nur kostenlose oder bereits vorhandene Software, sowie Hardware genutzt. Als Arbeitsgerät

wurde von der BREDEX GmbH ein Laptop mit Windows 10 und der Entwicklungsumgebung VisualStudio Code bereitgestellt. Des Weiteren wurde Firebase mit Firestore als Datenbank und Git als Versionsverwaltung eingesetzt.

2.4 Kostenplanung

Für den Prüfungsteilnehmer wird ein Stundenlohn von 9 EUR angesetzt. Ihm stehen für die Bearbeitung 80 Stunden zur Verfügung. Der Projektbetreuer führt das Anforderungsgespräch durch und steht dazu noch für weitere Fragen während der Bearbeitung zur Verfügung. Daher werden für ihn 6 Stunden angesetzt, bei einem geschätzten Stundenlohn von 45 EUR. Da nur bereits vorhandene oder kostenlose Ressourcen genutzt werden, ergeben sich keine weiteren Lizenz- oder Nutzungskosten.

Als Gesamtkosten berechnen sich hiermit 990 EUR (Tabelle 2).

| Akteur | Stundensatz (in EUR) | Dauer (in Stunden) | Kosten (in EUR) |
|---------------------|-------------------------|-----------------------|--------------------|
| Prüfungsteilnehmer | 9 | 80 | 720 |
| Projektbetreuer | 45 | 6 | 270 |
| Gesamtkosten | | | 990 |

Tabelle 2: Kostenplan

2.5 Nutzwertanalyse

Der Sinn der Anwendung liegt darin, den Prozess der Einarbeitung zu erleichtern. Unter diesem Gesichtspunkt wird der alte Prozess mit dem neuen Prozess, der die Anwendung nutzt, verglichen.

Am wichtigsten ist hierbei die Benutzerfreundlichkeit (Usability). Im alten Prozess mussten alle Aufgaben per Hand aus einer Word-Datei in eine eigene Liste kopiert werden und bei Erfüllung manuell durch Emojis abgehakt oder aus der eigenen Liste gelöscht werden. Außerdem mussten alle Aufgaben für nicht-deutschsprachige Mitarbeiter*innen vom Buddy selbst übersetzt werden.

In der Anwendung sind die Aufgaben über die Achievements direkt in einer Liste gesammelt und können durch einen einfachen Klick abgehakt werden. Die Sprache kann über einen Button in der Anwendung einfach umgestellt werden.

Der nächste Punkt ist die Einheitlichkeit. Derzeit gibt es unterschiedliche Listen mit verschiedenen Aufgaben. Je nach dem, welche Liste der Buddy nutzt, werden unterschiedliche Aufgaben erledigt.

Mit der Anwendung gibt es eine gesammelte Liste an Achievements, die von den Buddys genutzt wird. Zusätzlich kann diese Liste jederzeit von zuständigen Personen angepasst werden, sollte es Änderungen für die Einarbeitung geben.

Die Motivation diese Aufgaben zu erledigen steigt mit der Anwendung deutlich an. Der Vorgang, die Aufgaben zu suchen, und alles manuell kopieren zu müssen ist umständlich, was die Motivation senkt, sich darum zu kümmern. Mit der Anwendung müssen die Aufgaben nicht mehr gesucht werden, sondern sind einfach in der Applikation einsehbar. Außerdem erhält die Einarbeitung durch die Rangliste eine Gamification. Dadurch wird die Motivation gesteigert, da man sich mit seinen Mitarbeiter*innen messen kann und ein kleiner Wettbewerb entsteht.

Zuletzt wird noch die Sicherheit betrachtet. Die alten Dateien an Aufgaben lagen in zentralem Cloudspeicher und konnten von allem Mitarbeiter*innen bearbeitet werden oder neu erstellt werden. Mit der neuen Anwendung können nur noch zuständige Personen auf die Liste direkt zugreifen und diese bearbeiten.

| Kriterium | Gewichtung | Alte Struktur | | Neue Struktur | |
|------------------------|------------|---------------|----------|---------------|----------|
| | | Erfüllung | Nutzwert | Erfüllung | Nutzwert |
| Usability | 50% | 2 | 1 | 4 | 2 |
| Einheitlichkeit | 30% | 2 | 0,6 | 4 | 1,2 |
| Motivation | 10% | 1 | 0,1 | 4 | 0,4 |
| Sicherheit | 10% | 2 | 0,2 | 4 | 0,4 |
| SUMME | 100% | 7 | 1,9 | 15 | 4,0 |

Bewertungsmaßstab: 5 = sehr hoch, 4 = hoch, 3 = mittel, 2 = gering, 1 = sehr gering

Tabelle 3: Nutzwertanalyse

3 Analysephase

In der Analysephase werden die Anforderungen definiert, sowie der IST-Zustand analysiert und das SOLL-Konzept entwickelt. Damit kann im Anschluss die Architektur der Anwendung erstellt werden.

3.1 Anforderungsgespräch

Als Erstes wurde ein Anforderungsgespräch mit dem Kunden durchgeführt. In dem Gespräch wird der IST-Zustand ermittelt und das SOLL-Konzept definiert. Außerdem

wurden die vom Kunden gewünschten Features der Anwendung übermittelt. Die wichtigsten Features der Anwendung sind:

- Achievements, die abgehakt werden können
- Rangliste, um sich mit Mitarbeiter*innen vergleichen zu können
- Als Administrator Achievementliste bearbeiten können
- Mehrsprachigkeit in Deutsch und Englisch

Die Punkte wurden in dem Gespräch noch weiter konkretisiert, um die Anwendung im Anschluss möglichst nach den Wünschen des Kunden implementieren zu können.

3.2 IST-Zustand Analyse

Damit neue Mitarbeiter*innen der BREDEX GmbH besser in das Unternehmen finden, werden ihnen ein Ansprechpartner, den sogenannten BREDEX-Buddy an die Seite gestellt. Dieser soll dem*der Mitarbeiter*in das Unternehmen näherbringen. Allerdings ist nicht genau klar, was die Aufgaben des Buddys alles beinhalten. Es gibt unterschiedlichen Orten, an denen man Listen mit verschiedenen Inhalten findet. Dadurch führen die Buddys ihre Aufgabe unterschiedlich aus. Daraus resultiert, dass Mitarbeiter*innen unterschiedlich eingearbeitet werden und es besteht die Möglichkeit, dass Punkte vergessen oder vernachlässigt werden.

Zudem ist der Prozess sehr umständlich. Die Aufgaben müssen aus der Datei in eine Eigene kopiert werden, um sie abhaken zu können. Das Abhaken der erfolgt händisch über Löschen der Einträge, Einfügen eines Emojis oder Ähnlichem.

3.3 SOLL-Konzept

Um in Zukunft dafür zu sorgen, dass alle Buddys ihre Aufgaben auf die gleiche Art und Weise zu erledigen und des Prozess im Allgemeinen zu vereinfachen, soll eine Webanwendung programmiert werden. In der Anwendung soll es eine Checkliste an Achievements geben, die neue Mitarbeiter*innen im Verlauf ihrer Einarbeitung mit ihren Buddys zusammen abarbeiten können.

Zudem soll es eine Rangliste geben, in der sich Mitarbeiter*innen, die zur selben Zeit angefangen haben, gegenseitig sehen können. Diese Gamification soll die Motivation erhöhen, die Achievements zu erledigen, wodurch die neuen Mitarbeiter*innen sich mehr in das Unternehmen eingliedern.

Um die Achievements aktualisieren zu können, soll es noch eine Seite geben, bei der ausgewählte Personen neue Achievements anlegen, bearbeiten oder auch löschen können. Andere Personen sollen diese Seite weder sehen, noch darauf zugreifen können.

4 Designphase

In der Designphase werden Benutzeroberfläche und Datenmodelle der Anwendung geplant, um im vorhinein Schwierigkeiten oder Fehler erkennen zu können.

4.1 Benutzeroberfläche

Bei dem Design der Benutzeroberfläche (im folgenden GUI genannt) wird besonders darauf geachtet, dass alle Daten möglichst übersichtlich dargestellt werden. Dazu werden für jede der großen Anforderungen jeweils einzelne Unterseiten (im folgenden View genannt) unterteilt.

Hierbei gibt es eine View, um alle Achievements einsehen zu können und diese abhaken zu können. Achievements werden zu verschiedenen Zeitpunkten freigeschaltet. Manche sind direkt verfügbar, manche nach einer Woche, einem Monat oder erst nach einem halben Jahr. Entsprechend dieser Zeitstempel sind die Achievements in Akkordeons gruppiert. Dies führt dazu, dass Achievements übersichtlich einsehbar sind.

Eine weitere View gibt es für die Ansicht der Rangliste. Hier wird der derzeit eingeloggtten Person eine Rangliste mit den Personen die zu einer ähnlichen Zeit angefangen habe angezeigt, mit Name, Email und Punktzahl.

Eine letzte View gibt es, um die Achievements verwalten zu können. Die View ist ähnlich aufgebaut, wie die Achievements-View. Die Achievements werden in Akkordeons gruppiert, jedoch befindet sich neben jedem Titel der Akkordeons ein weiterer Button, um neue Achievements für den Zeitpunkt hinzufügen zu können. Außerdem ist der Button um Achievements abzuhaken ausgetauscht durch jeweils einen Button um das Achievement löschen oder bearbeiten zu können.

Beim Hinzufügen oder Bearbeiten eines Achievements öffnet sich ein Modaldialog, bei dem sich die Daten bearbeiten lassen und dann der Vorgang abgeschlossen werden kann. Hierbei werden fehlende Eingaben durch eine rote Umrandung des Eingabefeldes vermittelt.

4.2 Datenbank

Bei der Datenbank handelt es sich um die NoSQL Datenbank Firestore von Firebase. Die Daten werden im JSON-Format gespeichert. Hierbei wird von Sammlungen und Attributen gesprochen. Sammlungen sind mehrerer JSON-Objekte, mit einer ID und einem Datensatz. Diese Datensätze können aus weiteren Sammlungen oder Attributen wie Zahlen, Arrays oder ähnlichem bestehen.

Um die Achievements und die Daten der Nutzer der Anwendung speichern zu können wurden in der Datenbank jeweils Sammlungen angelegt.

5 Implementierungsphase

In der Implementierungsphase wird die Anwendung programmiert. Hierbei wird sich an den, in der Planung erstellten, Strukturen und Modellen bedient, um die Software möglichst ohne Komplikationen umsetzen zu können.

5.1 Frontend

Im Frontend wird das Webframework Vue eingesetzt. Dieses baut auf den Standard Websprachen HTML, JavaScript und CSS auf. Vue nutzt komponentenbasierte Programmierung, indem es HTML um die Möglichkeit erweitert, Templates zu erzeugen. Diese Templates werden dann entsprechend dem zugrunde liegenden Java-Scripts die GUI erzeugen. In dieser Anwendung wird Vue-Routing genutzt um eine Single-Page Anwendung zu erstellen.

Standardmäßig wird in Vue die Scriptsprache JavaScript genutzt. Das Framework kann auch mit der Sprache TypeScript genutzt werden, wofür sich in diesem Projekt entschieden wurde, um die Vorteile von Typisierung nutzen zu können. Dadurch können gewisse Fehler bereits zur Kompilierzeit vom Compiler entdeckt werden, wodurch die Anwendung weniger Fehleranfällig wird.

Vue stellt seit Version 3.0 auch eine neue API zur Verfügung: die Composition API. Die Vorteile gegenüber der, in Vue Standardmäßig genutzten, Options API, ist die Möglichkeit den Code variabler und besser zu strukturieren. Dies verbessert die Lesbarkeit und Qualität des Codes. Außerdem erleichtert die Composition API die Wiederverwendbarkeit von Code in verschiedenen Komponenten, was eine große Auswirkung auf Codequalität und Fehleranfällig hat.

Für Styling wird das CSS-Framework Bootstrap genutzt, um einfacher ein modernes Design für die Anwendung erstellen zu können und sich auf die Programmierung des Projektes konzentrieren zu können.

Als erstes wurde die Achievement-View erstellt, da es sich bei der View um die Hauptanforderung der Website handelt. Hier werden alle Achievements angezeigt, die der Benutzer erledigen kann und ob diese bereits erledigt wurden. Im oberen Teil der Website gibt es eine kurze Einführung und eine Fortschrittsleiste zeigt an, wie viel der Nutzer bereits geschafft hat. Jede Vue-Komponente hat seinen eigenen Lifecycle. Hierzu gehören beispielsweise das Erstellen, Mounten, Aktualisieren oder Unmounten der Komponente. Der OnMounted-Lifecycle Hook wird hier genutzt, um kurz vor dem Anzeigen der Komponente, die Daten aus der Datenbank zu laden. Diese Vorgehensweise wird in anderen Komponenten ebenfalls genutzt, um Daten zu laden.

Als nächstes wird die Login-View erstellt. Da in der BREDEX GmbH Microsoft Accounts genutzt werden, kann sich der Nutzer hier mit diesem Anmelden. Außerdem werden neue Nutzer erkannt, indem UUIDs verglichen werden und für jeweilige Nutzer neue Einträge in der Datenbank erstellt.

Im Anschluss wurde die Rangliste erstellt. Hier werden Mitarbeiter*innen, die bis zu einem Monat vor oder nach einem angefangen haben Tabellenartig mit Name, Punktzahl dargestellt.

Die letzte View ist zum Verwalten der Achievements. Auf diese View sollen nur Personen mit bestimmten Gruppen Zugriff haben, wie beispielsweise die Personalabteilung. Dieser Zugriff kann mit dem Vue-Router implementiert werden, welcher eine Möglichkeit hat, eine Bedingung an die Weiterleitung an die Adresse zu binden. In dieser Bedingung wird die Rolle des Nutzers abgefragt. Da diese bei der Anmeldung über Firebase nicht mit dem Access-Token übergeben wird, muss sich separat mit dem Azure Portal verbunden werden, um diese abzufragen.

Um zuletzt die Mehrsprachigkeit umzusetzen, wird sich das Vue Plugin Vue I18N zunutze gemacht. In einer separaten Datei wird zu jedem Text der Anwendung im JSON-Format ein Key gespeichert und die jeweiligen Übersetzungen. In dem HTML-Template wird nun anstelle des Textes der Key eingetragen, welcher beim Erstellen der Komponente, je nach aktuell eingestellter Sprache, durch die Übersetzung ersetzt wird.

Um die Anwendung bei Änderungen von Werten automatisch zu aktualisieren, gibt es in Vue ein Attribut, das eine Variable erhalten kann: das *Ref*. Vue rendert das DOM, bevor es auf dem Bildschirm dargestellt wird. Dadurch werden im Nachhinein veränderte Werte nicht mehr dagesetellt. Refs haben hierbei einen Sonderstatus. Sobald sich die Variable mit dem Attribut ändert, wird die Komponente mit dem neuen Wert neu berechnet.

5.2 Backend

Für das Backend wird das Backend-As-A-Service Firebase genutzt. Darin sind verschiedene Optionen zur Benutzeranmeldung, wovon nur die Anmeldung über Micorsoft genutzt wird, sowie die NoSQL Datenbank Firestore. Als erstes muss das Backend mit der Software verbunden werden. Dafür muss als erstes die Firebase SDK mit dem Befehl

```
npm install firebase
```

über Node.js installiert werden und im Anschluss in den Abhängigkeiten der Vue-Anwendung hinzugefügt werden. Nun muss nur noch die Vue-App mit den Firebase Einstellungen konfiguriert werden.

Als nächstes wird die Datenbank erstellt. Die NoSQL Datenbank basiert auf JSON-ähnlichen Dateistrukturen. Man kann Sammlungen erstellen, mit verschiedenen Elementen. Diese Elements haben jeweils eine ID, welche in einer Zeichenkette gespeichert wird. Es wird eine Sammlung erstellt für die Nutzer und eine Sammlung für die Achievements.

Die Achievements müssen für beidem Sprachen ihren Text enthalten. Diese werden jeweils in einer Zeichenkette gespeichert. Des weiteren muss der Zeitpunkt gespeichert werden, in dem das Achievement freigeschaltet werden. Da diese an

vier verschiedenen möglichen Zeitpunkten freigeschaltet werden können, wurde sich dazu entschieden, das Attribut in einem Wert von 0 bis 3 zu speichern. Dadurch können die Achievements leicht gruppiert und mit einem Zähler iteriert werden. Die Punkte, die jedes Achievement hat, werden in einer Ganzzahl gespeichert.

Für die ID des Nutzers, wird beim Anlegen diesen, die Microsoft UUID genutzt. So kann nach der Anmeldung leicht wieder auf die Daten zugegriffen werden. Abgesehen von der ID wird für den Nutzer der Zeitpunkt gespeichert, an der sich der Nutzer das erste mal angemeldet hat. Dadurch kann beim Abruf der Nutzer für die Rangliste bereits in der Datenbank gefiltert werden. Weiterhin werden noch Stammdaten wie Name oder Email-Adresse gespeichert. Zuletzt wird für jeden Nutzer noch eine Liste an Zeichenketten gespeichert. Diese Liste beinhaltet alle IDs der Achievements, die der Nutzer bereits erledigt hat. Über diese IDs kann sich effizient die Liste an erledigten Achievements geholt werden, um Punkte oder die prozentuelle Anzahl erledigter Achievements zu berechnen.

6 Testphase

Um die korrekte Arbeitsweise der Anwendung zu garantieren, werden während der Entwicklung regelmäßige Tests durchgeführt. Mithilfe von Unit-Tests wurde während der Entwicklungsphase sichergestellt, dass Methoden korrekt funktionieren. Durch exploratives Testen wurde außerdem sichergestellt, dass die Anwendung bei Eingaben das erwartete Verhalten erfüllt und keine ungültigen Eingaben getätigt werden können.

Da bei der Software die Usability im Vordergrund steht, wurden zusätzlich User-tests durchgeführt. Außerdem gab es Code Reviews von erfahreneren Entwicklern, um Codequalität sicherzustellen, oder effizientere Wege gezeigt zu bekommen, um diese im Anschluss anwenden zu können.

7 Fazit

7.1 Soll/Ist-Vergleich

Während der Durchführung der Projektarbeit kam es zu kleinen Differenzen zu der initialen Zeitplanung.

Da ein paar Anforderungen während der Implementierungsphase hinzu kamen, musste in die Designphase zurückgewechselt werden, da das Design erst überarbeitet werden musste, um den neuen Anforderungen gerecht zu werden. Dieser Aufwand kostete den Prüfungsteilnehmer in etwa eine Stunde. Auch in der Implementierungsphase kam es durch wenige Kenntnisse in der Webentwicklung und über das Vue Framework zu einigen Verzögerungen. Die dadurch entstandenen

Verzögerungen konnten jedoch durch gut gewählte Bibliotheken wieder aufgeholt werden und zudem die verlorene Zeit aus der Designphase ausgeglichen werden.

In den restlichen Phasen kam es zu keinen weiteren Abweichungen.
Ein graphischen Vergleich der Änderungen der Zeitplanung bietet Tabelle 4.

| Phase | geplante Dauer (in Stunden) | tatsächliche Dauer (in Stunden) | Differenz (in Stunden) |
|----------------------|--------------------------------|------------------------------------|---------------------------|
| Anforderungsgespräch | 3 | 3 | 0 |
| Analysephase | 6 | 6 | 0 |
| Planungsphase | 6 | 6 | 0 |
| Designphase | 6 | 7 | +1 |
| Umsetzung | 40 | 39 | -1 |
| Test und Abnahme | 7 | 7 | 0 |
| Dokumentation | 12 | 12 | 0 |
| Gesamt | 80 | 80 | +0 |

Tabelle 4: Zeitplanung; Differenzen

7.2 Lessons learned

Während des Projektes lernte der Prüfungsteilnehmer sehr viel über Webentwicklung und das Erstellen und Umsetzen von Projekten. Vor allem die Wichtigkeit und Auswirkung einer guten Design und Analysephase in einem Projekt haben sich gezeigt. Aber auch die Wahl von Technologie und genutzten Bibliotheken können einen großen Einfluss auf den flüssigen und schnellen Ablauf der Softwareentwicklung haben.