



# Projektantrag LF12

„Entwicklung eines On-Premise Chat-Tools“

Von: Ziggy, Franz, Rouven und Elmo

# 1. Projektbezeichnung

Prototypenentwicklung eines On-Premise Chat-Tool für eine nach außen abgesicherte Kommunikation im Hinblick auf besonders kritische Daten.

## 1.1 Kurzform der Aufgabenstellung

Die ZFRE IT-Solutions GmbH wurde damit beauftragt einen Prototyp für die On-Premise Kommunikation eines Kunden zu entwickeln. Dieses Tool soll im Aufbau und in der Funktionsweise Tools wie Slack und Teams ähneln, allerdings im lokalen Netz des Kunden bereitgestellt werden und keine Verbindung nach außen aufbauen.

## 1.2 Ist Analyse

Der Kunde arbeitet regelmäßig mit sensiblen Daten, die unter die Verschlusssachenanweisung (VSA) fallen und einen Geheimhaltungsgrad von VS-VERTRAULICH bis zu STRENG GEHEIM haben. Aufgrund der kritischen Natur dieser Daten ist die digitale Infrastruktur beim Kunden streng gekapselt und es sind strenge Zugriffskontrollen eingerichtet worden.

Was sich allerdings kaum vermeiden lässt ist interne Kommunikation über diese Dokumente. In den täglichen Geschäftsprozessen des Kunden ist es wichtig, dass verschiedene Mitarbeiter sich über ihre Arbeit abstimmen können. Im Bezug auf diese Art der Kommunikation ist es zwar nicht gestattet den Namen eines solchen Dokumentes zu verwenden (stattdessen werden Codewörter genutzt), allerdings dürfen Mitarbeiter Aussagen über den Status ihrer Arbeit an solchen Dokumenten treffen.

Bei diesem Prozess wurde bisher ein interner Mail-Server genutzt, der eine On-Premise Lösung für dieses Problem darstellt. Allerdings hat der Mail-Verkehr aufgrund einer höheren Auftragslage in letzter Zeit stark zugenommen und viele Mitarbeiter empfinden die Kommunikation per Mail vor allem mit mehreren Beteiligten als unübersichtlich. Die Nutzung eines industrieeüblichen Tools wie Slack oder Teams ist nicht umsetzbar, da selbst das Versenden von Telemetriedaten eine potenzielle Lücke in der Cybersicherheit des Kunden darstellt.

# 2. Zielsetzung entwickeln

## 2.1 Was soll am Ende des Projektes erreicht sein?

Der Kunde wünscht einen einfachen Prototyp, um sein Sicherheitskonzept zu validieren. Dieser Prototyp soll dabei im lokalen Netz des Kunden laufen können und soll keine externe Verbindung aufbauen. Der Funktionsumfang des Prototyps soll aus den wesentlichen Aspekten von bekannten Chat-Tools bestehen, spezifisch soll es möglich sein die direkte Kommunikation zwischen zwei Nutzern

und die Kommunikation von einer Gruppe an Nutzern untereinander, so zu ermöglichen, dass alle Daten, die dabei generiert werden, On-Premises bleiben. Ebenfalls ist es dem Kunden wichtig, dass Administratoren alle Gespräche überwachen können. Das Tool soll als Webseite im lokalen Netz verfügbar sein und über die vom Kunden eingesetzten Browser (Firefox und Chrome) erreichbar sein.

## 2.2 Welche Anforderungen müssen erfüllt sein?

Das Chat-Tool muss folgende Anforderungen erfüllen:

- Ein intuitives Design
- Die Möglichkeit sich als Nutzer anzumelden
- Nur Administratoren sollen Nutzer anlegen können
- Direkte Chats
- Gruppen Chats
- Administratoren müssen alle Chats beobachten können
- Es muss eine Liste an verbotenen Worten geben, dessen Verwendung in einer Nachricht dazu führt, dass diese Nachricht nicht versandt wird

## 2.3. Welche Einschränkungen müssen berücksichtigt werden?

Durch das strenge Sicherheitskonzept wird es als kritisch angesehen, dass in dem Quellcode der Software auf unnötige Abhängigkeiten verzichtet wird. Ebenfalls wünscht der Kunde, dass als Programmiersprache JavaScript (auch TypeScript) verwendet wird, da das Entwicklerteam des Kunden dort die meisten Erfahrungen hat und eine Übernahme der Entwicklung durch den Kunden in Zukunft offenbleiben soll. Ebenfalls soll das Tool TLS in der Kommunikation von Client zu Server einsetzen.

# 3. Projektstrukturplan erstellen

## 3.1 Was ist zur Erfüllung der Zielsetzung erforderlich?

Das Projekt soll in einem erweiterten Wasserfallmodell entwickelt werden. Durch die präzisen Anforderungen eignet sich dieses Modell besonders für dieses Projekt. Sollte es in einer der Projektphasen unerwartet zu Problemen kommen, kann in eine vorherige Projektphase gesprungen werden, um dort eine Lösung zu finden. Dieses Vorgehen ermöglicht es die klaren Anforderungen für eine schnelle und gezielte Entwicklung zu nutzen, ohne dass ein Mehraufwand für beständige Kommunikation mit dem Kunden besteht.

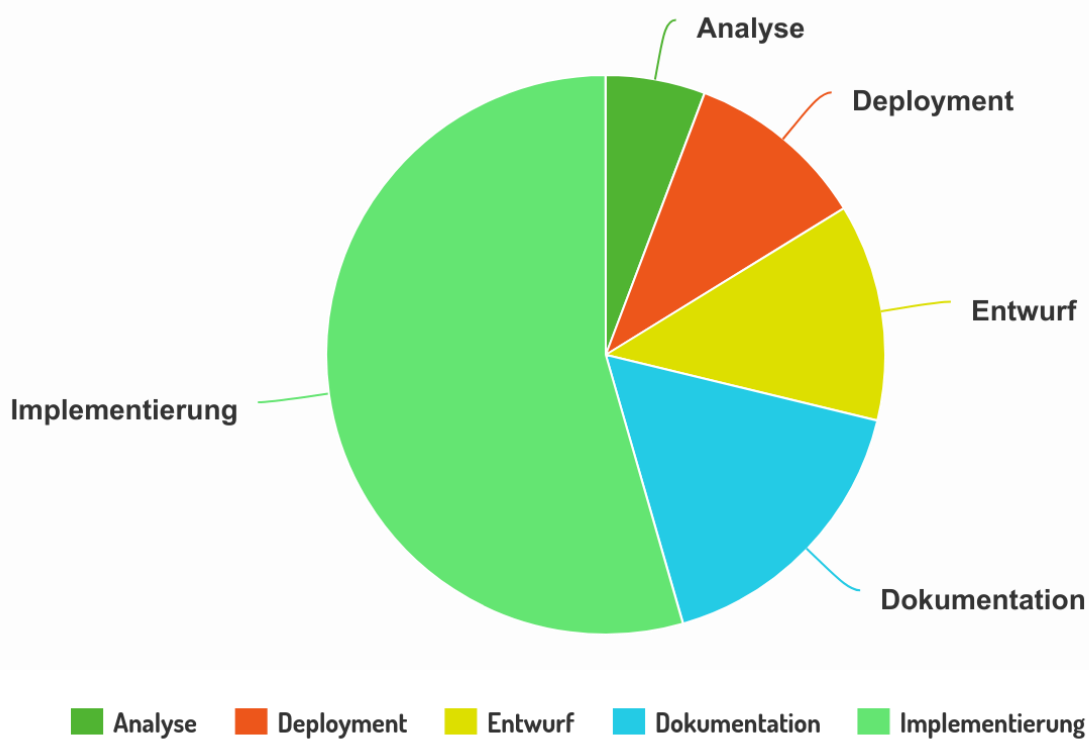
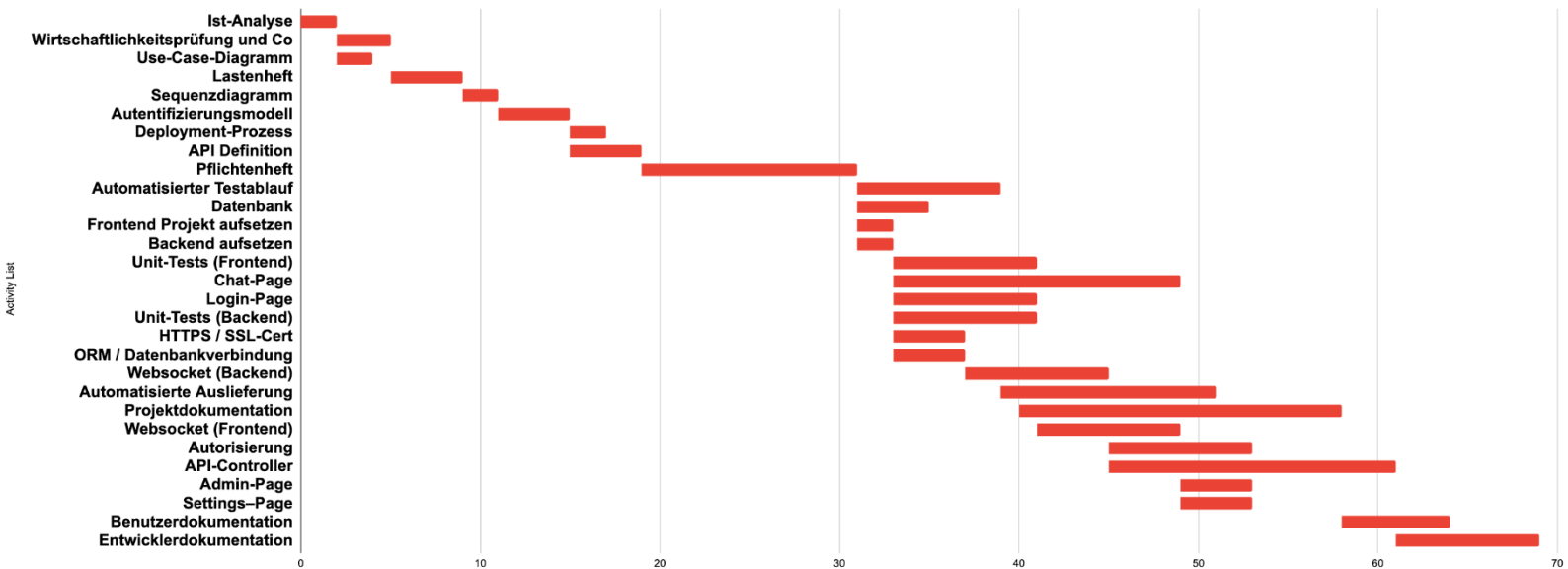
Das Projekt wird in Git versioniert und durchgehend über eine Pipeline mit Unit-Tests getestet. Das gesamte Projekt wird in drei Teilen dokumentiert, einmal in einer Projektdokumentation, die den gesamten Projektablauf festhält, dann in einer Entwicklerdokumentation, die aus Sicht eines Entwicklers den Quellcode und

den Programmablauf beschreibt und zuletzt in einer Anwenderdokumentation, die den eigentlichen Endnutzern des Tools, die genaue Bedienung erklärt.

## 3.2 Aufgaben auflisten

- Analyse
  - Ist-Analyse erweitern und ausarbeiten
  - Use-Case-Diagramm
  - Wirtschaftlichkeitsprüfung und Amortisationsrechnung des Projektes durchführen
  - Erstellung des Lastenheftes
- Entwurf
  - Sequenzdiagramm für die Server zu Client Kommunikation
  - Authentifizierungsmodell aussuchen
  - API-Definition aufstellen
  - Deployment-Prozess entwickeln
  - Pflichtenheft erstellen
- Implementierung
  - Backend
    - Backend Projekt aufsetzen
    - ORM / Datenbankverbindung aufbauen
    - Websocket implementieren
    - API-Controller implementieren
    - HTTPS / SSL-Cert
    - Autorisierung implementieren
    - Unit-Tests implementieren
  - Frontend
    - Frontend Projekt aufsetzen
    - Websocket implementieren
    - Login-Page
    - Chat-Page
    - Settings-Page
    - Admin-Page
    - Unit-Tests implementieren
  - Datenbank
    - Datenbank einrichten
- Deployment
  - Automatisierter Testablauf
  - Automatisierte Auslieferung des Projektes
- Dokumentation
  - Erstellen der Projektdokumentation
  - Erstellen der Entwicklerdokumentation
  - Erstellen der Benutzerdokumentation

### 3.3 Grafische und tabellarische Darstellung



meta-chart.com

Phase	Dauer in Stunden
Analyse	11
Entwurf	24
Implementierung	104
Deployment	20
Dokumentation	32

Summe:	191
--------	-----

## 4. Projektphasen mit Zeitplanung in Stunden

<b>Analyse</b>	<b>11 h</b>
Ist-Analyse erweitern und ausarbeiten	2 h
Use-Case-Diagramm	2 h
Wirtschaftlichkeitsprüfung und Amortisationsrechnung des Projektes durchführen	3 h
Erstellung des Lastenheftes	4 h
<b>Entwurf</b>	<b>24h</b>
Sequenzdiagramm für die Server zu Client Kommunikation	2 h
Authentifizierungsmodell aussuchen	4 h
API-Definition aufstellen	4 h
Deployment-Prozess entwickeln	2 h
Pflichtenheft erstellen	12 h
<b>Implementierung</b>	<b>104 h</b>
Backend Projekt aufsetzen	2 h
ORM / Datenbankverbindung aufbauen	4 h
Websocket implementieren	8 h
API-Controller implementieren	16 h
HTTPS / SSL-Cert	4 h
Autorisierung implementieren	8 h
Unit-Tests implementieren	8 h
Frontend Projekt aufsetzen	2 h
Websocket implementieren	8 h
Login-Page	8 h
Chat-Page	16 h
Settings-Page	4 h
Admin-Page	4 h
Unit-Tests implementieren	8 h
Datenbank einrichten	4 h
<b>Deployment</b>	<b>20 h</b>

Automatisierter Testablauf	8 h
Automatisierte Auslieferung des Projektes	12 h
<b>Dokumentation</b>	<b>32 h</b>
Erstellen der Projektdokumentation	18h
Erstellen der Entwicklerdokumentation	8h
Erstellen der Benutzerdokumentation	6h