### Testpläne und Teststrategien

| **Feld** | **Beschreibung** |
| --- | --- |
| **Name des Artefakts** | Testpläne und Teststrategien |
| **Verantwortlich** | IT-Systemverantwortlicher, Software-Entwickler |
| **Ziel des Dokuments** | Festlegung der Vorgehensweise und Methoden zur Durchführung von Tests zur Qualitätssicherung. |
| **Wesentliche Inhalte** | - Teststrategien  - Testpläne  - Testmethoden und -werkzeuge  - Testumgebungen |
| **Erstellungszeitpunkt** | Anfangsphase der Entwicklungsphase des Projekts, nach Erstellung der technischen Spezifikationen |
| **Nutzung im Projektverlauf** | Grundlage für die Durchführung von Tests und die Qualitätssicherung der Anwendung |
| **Abnahme** | Durch die IT-Systemverantwortlichen und die Entwickler; Überprüfung der Vollständigkeit und Korrektheit der Testpläne und Teststrategien |
| **Abhängigkeiten** | - Technische Spezifikationen  - Anwendungsfälle |
| **Versionierung** | Versionskontrolle mittels Versionsmanagement-Tools |
| **Vorlagen und Tools** | Testmanagement-Tools (z.B. JIRA, TestRail) |
| **Zusätzliche Anmerkungen** | Regelmäßige Überprüfung und Aktualisierung der Testpläne und Teststrategien notwendig, um sicherzustellen, dass sie den aktuellen Anforderungen entsprechen und vollständig sind |

# Testpläne und Teststrategien für das CRM-Projekt

## 1. Einleitung

Dieses Dokument beschreibt die Testpläne und Teststrategien für das CRM-Projekt, das um die Befüllung mit Scoring-Daten erweitert wurde. Ziel ist es, die Qualität und Funktionalität der Erweiterung sicherzustellen und potenzielle Fehler und Sicherheitslücken zu identifizieren.

## 2. Ziele des Testplans

* Sicherstellung der Funktionalität der neuen Scoring-Daten-Upload-Funktion
* Validierung der Datenintegrität und -konsistenz
* Überprüfung der Benutzerfreundlichkeit und Performance
* Gewährleistung der Sicherheit und des Datenschutzes

## 3. Testumfang

Der Testumfang umfasst die folgenden Komponenten des CRM-Systems:

* Upload-Interface für Scoring-Daten
* Validierungslogik für hochgeladene Daten
* Datenbankintegration für Scoring-Daten
* API-Endpunkte für das Abrufen und Aktualisieren von Scoring-Daten
* Benutzeroberflächenanpassungen

## 4. Teststrategien

### 4.1 Funktionale Tests

**Ziel:** Sicherstellen, dass alle neuen und bestehenden Funktionen wie erwartet arbeiten.

* **Testfälle:**
  + Hochladen einer gültigen Excel-Datei
  + Hochladen einer ungültigen Excel-Datei
  + Automatische Zuordnung der Scoring-Daten zu den Kundendatensätzen
  + Anzeige der Scoring-Werte im Kundenprofil
  + API-Endpunkte: POST /upload, GET /customers/:id/score, PUT /customers/:id/score

### 4.2 Nicht-funktionale Tests

**Ziel:** Überprüfung der nicht-funktionalen Anforderungen wie Leistung, Skalierbarkeit und Benutzerfreundlichkeit.

* **Testfälle:**
  + Performance-Test des Upload-Prozesses bei großen Datenmengen
  + Lasttest bei gleichzeitigen Uploads
  + Usability-Test der Benutzeroberfläche für den Scoring-Daten-Upload

### 4.3 Regressionstests

**Ziel:** Sicherstellen, dass die bestehenden Funktionen nach der Integration der neuen Features weiterhin einwandfrei funktionieren.

* **Testfälle:**
  + Überprüfung der bestehenden Kundenverwaltungsfunktionen
  + Überprüfung der Datenbankoperationen für Kundendaten
  + Validierung der bestehenden API-Endpunkte

### 4.4 Sicherheitstests

**Ziel:** Sicherstellen, dass die neuen Funktionen sicher sind und keine Sicherheitslücken einführen.

* **Testfälle:**
  + Prüfung der Authentifizierung und Autorisierung für API-Endpunkte
  + Überprüfung der Verschlüsselung der hochgeladenen Daten
  + Penetrationstests auf mögliche Schwachstellen

### 4.5 Usability-Tests

**Ziel:** Sicherstellen, dass die Benutzeroberfläche intuitiv und benutzerfreundlich ist.

* **Testfälle:**
  + Überprüfung der Benutzerführung beim Hochladen der Scoring-Daten
  + Feedback der Benutzer zur Benutzerfreundlichkeit der neuen Funktionen

## 5. Testumgebung

* **Hardware:** Server mit ausreichender Leistung zur Verarbeitung der Testdaten
* **Software:** Testumgebungen für Web-Frontend, Backend und Datenbank
* **Daten:** Testdatenbank mit anonymisierten Kundendaten

## 6. Testdaten

* **Testdaten für Scoring-Uploads:** Beispiel-Excel-Dateien mit gültigen und ungültigen Daten
* **Benutzerkonten:** Testbenutzer mit verschiedenen Berechtigungen (Admin, normaler Benutzer)

## 7. Testfälle und Szenarien

### Beispiel-Testfälle:

1. **Hochladen einer gültigen Excel-Datei**
   * **Beschreibung:** Eine Excel-Datei mit korrekten Daten wird hochgeladen.
   * **Erwartetes Ergebnis:** Die Datei wird erfolgreich hochgeladen und die Daten werden korrekt zugeordnet.
2. **Hochladen einer ungültigen Excel-Datei**
   * **Beschreibung:** Eine Excel-Datei mit fehlenden oder fehlerhaften Daten wird hochgeladen.
   * **Erwartetes Ergebnis:** Die Datei wird abgelehnt und eine entsprechende Fehlermeldung wird angezeigt.
3. **API-Endpunkt GET /customers/:id/score**
   * **Beschreibung:** Abrufen des Scoring-Werts für einen vorhandenen Kunden.
   * **Erwartetes Ergebnis:** Der korrekte Scoring-Wert wird zurückgegeben.

## 8. Testwerkzeuge

* **Automatisierte Testwerkzeuge:** Selenium, PHPUnit
* **Manuelle Testwerkzeuge:** TestRail, JIRA für das Bug-Tracking
* **Sicherheitstest-Tools:** OWASP ZAP, Burp Suite

## 9. Rollen und Verantwortlichkeiten

* **Testmanager:** Verantwortlich für die Planung und Koordination der Tests.
* **Testentwickler:** Erstellung und Durchführung der Testfälle.
* **QA-Ingenieure:** Durchführung der manuellen und automatisierten Tests.
* **Sicherheitsanalysten:** Durchführung der Sicherheitstests.

## 10. Zeitplan

| **Phase** | **Dauer** | **Verantwortlich** |
| --- | --- | --- |
| Planung | 1 Woche | Testmanager |
| Vorbereitung der Testumgebung | 1 Woche | Testentwickler |
| Entwicklung der Testfälle | 2 Wochen | Testentwickler |
| Durchführung der Tests | 3 Wochen | QA-Ingenieure |
| Fehlerbehebung und Retests | 2 Wochen | Entwickler, QA-Ingenieure |
| Abschluss und Berichtserstellung | 1 Woche | Testmanager |

## 11. Risiken und Gegenmaßnahmen

| **Risiko** | **Gegenmaßnahme** |
| --- | --- |
| Verzögerungen bei der Testvorbereitung | Pufferzeiten einplanen, klare Prioritäten setzen |
| Unzureichende Testdaten | Testdaten frühzeitig vorbereiten, echte Daten anonymisieren |
| Sicherheitslücken werden übersehen | Durchführung regelmäßiger Penetrationstests, externe Audits |
| Fehlende Ressourcen | Zusätzliche Ressourcen einplanen, frühzeitige Kommunikation |

## 12. Abnahmekriterien

* **Funktionale Tests:** Alle funktionalen Testfälle müssen erfolgreich bestanden werden.
* **Nicht-funktionale Tests:** Die Systemleistung muss den Anforderungen entsprechen.
* **Sicherheitstests:** Es dürfen keine kritischen Sicherheitslücken vorhanden sein.
* **Usability-Tests:** Positive Rückmeldungen von den Benutzern.