**7. Wartungsinformationen**

**7.1 Entwicklungsannahmen**

Das System wird unter der Annahme entwickelt, dass:

* Die To-Do-Liste als plattformunabhängige mobile Applikation bereitgestellt wird (z. B. mithilfe von Flutter oder React Native).
* Die App auf den gängigen mobilen Betriebssystemen **Android** und **iOS** lauffähig ist.
* Eine serverseitige Architektur mit zentraler Datenbank verwendet wird, mit der die App über eine API kommuniziert.
* Nutzer:innen die App offline verwenden können, wobei eine Synchronisation mit dem Server erfolgt, sobald wieder eine Internetverbindung besteht.
* Die App möglichst intuitiv und ohne große Einarbeitung nutzbar sein soll.

**7.2 Erwartete Änderungen**

Im Verlauf der Weiterentwicklung sind folgende Änderungen wahrscheinlich:

* **Plattformoptimierungen**: Verbesserungen der Usability speziell für Android bzw. iOS.
* **Erweiterung der Synchronisationslogik**: Bessere Handhabung von Offline-Modus und Konfliktlösung.
* **Push-Benachrichtigungen**: Erinnerungen an fällige Aufgaben oder Team-Aktivitäten.
* **Erweiterung durch neue Features**: z. B. Sprachsteuerung, Integration mit Kalender-Apps oder Wearables.

**7.3 Änderungsfreundliche Komponenten**

Die App wird modular aufgebaut, um zukünftige Änderungen und Erweiterungen möglichst einfach zu machen:

* **Modulares Frontend**: Die Benutzeroberfläche basiert auf wiederverwendbaren Komponenten.
* **API-first-Ansatz**: Die App ist über eine RESTful-API mit dem Server verbunden, was eine lose Kopplung ermöglicht.
* **Lokal und Serverseitig gespeicherte Daten**: Die Architektur berücksichtigt sowohl lokale Speicherung (z. B. mit SQLite oder Shared Preferences) als auch Cloud-Synchronisation.

**7.4 Dokumentation und Versionierung**

* Der Quellcode wird versioniert (z. B. mit Git).
* Änderungen werden in Form von Tickets/Issues dokumentiert.
* Ein changelog wird gepflegt, insbesondere für neue Releases.
* Für größere Änderungen werden Architekturentscheidungen dokumentiert (z. B. in Form von ADRs).

**7.5 Deployment und Update-Strategien**

* **Plattformabhängiger Rollout**: Neue App-Versionen werden über den **Google Play Store** bzw. den **Apple App Store** bereitgestellt.
* **Release-Management**: Kleinere Updates (Bugfixes) erfolgen häufiger, größere Feature-Updates nach definierten Meilensteinen.
* **Beta-Testprogramm**: Über Google Play Console oder TestFlight können Vorabversionen an Testnutzer:innen verteilt werden.
* **Sicherheitsupdates und Hotfixes**: Kritische Fehler oder Sicherheitslücken werden bevorzugt behandelt und kurzfristig ausgerollt.