

Risiko	Beschreibung	Wahrscheinlichkeit	Schadenshöhe	Minimierung	Monitoring	Notfallplan	Status	verantwortliche	Datum
Unterschiedliche Programmierumgebungen	Einer Nutzt Visual Studio, der rest Jetbrains	Hoch	Mittel	Docker nutzen, .env, Versionsvorgaben	CI/CD Build prüfen	Einheitliche Dev-Container einführen	Behoben	Max	17.04.2025
Raspberry Pi fällt aus	Der Server/Raspberry Pi geht aus und der Server ist nicht mehr erreichbar	Sehr Niedrig	Mittel	Watchdogs, Redundantes System	dauerhaft anpingen	Backup-RPi bereitstellen	nicht kritisch	-	17.04.2025
Unerfahrenes Team	-	Hoch	Mittel	Aufgaben klein halten, Code Reviews	Meilensteine kontrollieren	Zeitpuffer einplanen, Lernphasen einbauen Technische Schulden	offen	-	17.04.2025
Schlechte Codequalität / Bugs	Einige Features könnten nicht oder nicht richtig funktionieren	Mittel	Hoch	Pylint nutzen, Tests	Fehler-Logs, GitHub Issues	dokumentieren & aufarbeiten	offen	-	17.04.2025
Sicherheitslücken im Messenger	Sicherheitslücken, sodass dieser nicht sicher ist.	Hoch	Hoch	HTTPS, Input-Validierung, Authentifizierung	Penetrationstest, Dependency-Scan	Kritische Lücken sofort patchen	offen	-	17.04.2025
Verzögerungen durch Uni/Privatleben	-	Hoch	Mittel	Pufferzeiten einplanen	Fortschrittsmeetings	Priorisierung, Aufgaben ggf. umverteilen	Behoben	-	17.04.2025
Backend/Frontend-Kommunikation bricht	Das Backend wurde Aktualisiert, das Frontend noch nicht.	Hoch	Hoch	Schnittstellenspezifikation	Alles gut absprechen	Klare Fehlerbehandlung & Reconnect-Logik	Minimiert	Max & Pascal	17.04.2025