Software Quality Assurance - Report

Übersicht

Projekt: Projekt Episko
Autor: Simon Blum
Datum: 02.04.2025
Zuletzt geändert:
von: Max Rodler
am: 03.04.2025
Version: 2
Prüfer: Paul Stöckle
Letzte Freigabe:
durch: Paul Stöckle
am: 03.04.2025

Changelog

Datum	Verfasser	Kurzbeschreibung
02.04.2025	Simon Blum	Initiales Erstellen und Verfassen
03.04.2025	Max Roder	Formatierung und Korrekturen

Distribution List

- Simon Blum simon21.blum@gmail.com
- Ben Oeckl ben@oeckl.com
- Maximilian Rodler maximilianreinerrodler@gmail.com
- Paul Stöckle paul.stoeckle@t-online.de

Zusammenfassung der Qualitätssicherungstätigkeiten

Überblick Reviews

Inkrement	Ergebnis	Kommentar
1	Abgenommen	Abweichendes Verfahren
2	Abgenommen	Abweichendes Verfahren
3	Abgenommen	Abweichendes Verfahren
4	Abgenommen	Abweichendes Verfahren
5	Abgenommen	Abweichendes Verfahren
6	Teilweise Abgenommen	2 Arbeitspakete verlängert

Inkrement	Ergebnis	Kommentar
7	Abgenommen	-
8	Abgenommen	-
9	Teilweise Abgenommen	Teils fehlende Dokumentation

Nach der Durchführung der Inkrement Reviews wurde festgestellt, dass in der Dokumentation noch mehr Informationen enthalten sein sollten. Es wird zudem darauf hingewiesen, dass bei jedem Inkrement Review manuelle Tests durchgeführt werden, um die Funktionalität und Stabilität des Systems zu gewährleisten. Zusätzlich werden vor jedem Merge CodeReviews vorgenommen, um den Abschluss eines jeden Inkrements abzusichern.

Konformität mit Software Quality Assurance Plan

Im Rahmen des Projekts wird die Codequalität automatisch mithilfe von Clippy und SvelteCheck in der CI-Pipeline geprüft. Zudem wird der Code in der Pipeline automatisch mit Prettier und rustfmt formatiert, um einen einheitlichen Stil sicherzustellen. Des Weiteren erfolgt die Testdurchführung automatisch in der Pipeline. Somit wurden alle im QA-Plan vorgesehenen Maßnahmen vollständig umgesetzt.

Ergebnisse der Verifikations- und Validierungsmaßnahmen

Beim manuellen Lasttest wurde zunächst festgestellt, dass die Nutzererfahrung bei einer hohen Anzahl von Manifesten unbefriedigend war. Diese Situation konnte jedoch durch die Einführung einer Pagination optimiert werden, sodass die Leistung und Benutzerfreundlichkeit verbessert wurden.

Audit- und Review-Ergebnisse

Festgestellte Abweichungen und nicht erfüllte Anforderungen.

Requirement	Status	Kommentar
FA1.2.2	Teilweise erfüllt	Es wird geprüft, Ergebnisse der Prüfung jedoch ignoriert
FA1.4.3	Nicht erfüllt	Keine Überprüfung bei Laden aus Konfiguration

Requirement	Status	Kommentar
FA2.2.3	Teilweise erfüllt	Gelöschte Dateien werden nicht als solche erkannt
FA3.1.1	Teilweise erfüllt	Daten können nur nach Titel durchsucht werden
NA4.1	Nicht erfüllt	Es wird nur ein ".deb" Artifact erstellt
NA10.1	Nicht erfüllt	Auf zukünftige Internationalisierung wurde keine Rücksicht genommen

Maßnahmen zur Behebung identifizierter Probleme.

Dank der modularen Architektur können die identifizierten Probleme relativ leicht behoben werden. Es wurden keine kritischen Probleme festgestellt, sodass anstehende Korrekturen in zukünftigen Releases fix umgesetzt werden können.

Qualitätsmetriken und Leistungsbewertung

Analyse der erfassten Qualitätsmetriken

Die Analyse der Qualitätsmetriken ergab, dass die C0 Coverage bei 48 % liegt, was unter dem angestrebten Zielwert liegt. Ebenso beträgt die C1 Coverage 56 %, was ebenfalls unter dem Zielwert liegt. Positiv hervorzuheben ist, dass sowohl durch Clippy als auch durch SvelteCheck keine Probleme festgestellt wurden.

Lessons Learned und Empfehlungen

Identifikation von Verbesserungspotenzialen für zukünftige Projekte.

Die retrospektive Analyse hat gezeigt, dass bestehende Prozesse stellenweise nicht ausreichend dokumentiert waren. Zukünftige Projekte sollten stärker auf frühzeitige Prozessanalyse und kontinuierliches Feedback setzen. Einzelne Schwachstellen im QA-Prozess, wie z. B. die verspätete Einführung strukturierter Tests, weisen auf Verbesserungspotenziale hin. Eine genauere Auswertung dieser Punkte ist für künftige Projekte sinnvoll.

Vorschläge zur Optimierung von Qualitätssicherungsprozessen.

Es wird empfohlen, bereits in früheren Projektphasen eine strukturierte Pipeline zu implementieren, um die Qualitätssicherungsprozesse weiter zu optimieren und zu beschleunigen.

Schlussfolgerungen und Freigabeempfehlungen

Gesamtbewertung der Softwarequalität.

Die Gesamtbewertung der Softwarequalität zeigt, dass die Codequalität als gut einzustufen ist und auch die zugrundeliegende Architektur den Anforderungen entspricht. Allerdings weist die Testabdeckung noch deutlichen Verbesserungsbedarf auf.

Empfehlung für die Freigabe oder notwendige Nachbesserungen.

Aufgrund der Tatsache, dass keine kritischen Probleme vorliegen und es sich um eine nicht kritische Anwendung handelt, kann die Freigabe erteilt werden. Zwar sind stellenweise Verbesserungen möglich, diese stellen jedoch keine Blockade für die Freigabe dar.