



AGENTIC CODING IN BESTANDSANWENDUNGEN

AUTOMATISIERTE, ARCHITEKTURKONFORME UMSETZUNG VON USER STORIES MIT RULES, TESTS UND REVIEW-LOOP

AUSGANGSSITUATION

Unser Kunde, eine international tätige Versicherungsgruppe mit starkem Industrie- und Firmenkundengeschäft, betreibt eine umfangreiche, gewachsene Anwendungslandschaft mit Frontend- und Backend-Komponenten. Ziel des Projekts war es, bestehende Anwendungen so zu „KI-ifizieren“, dass der Entwicklungsprozess von der JIRA-User-Story bis zum Code-Commit weitgehend automatisiert erfolgen kann. Die Lösung sollte ausschließlich mit Claude Code und Kiro realisiert werden, Architektur- und Programmierrichtlinien strikt einhalten und sowohl die Entwicklung als auch die Analyse von Produktionsproblemen unterstützen. Zentrale Anforderungen waren hohe Architekturkonformität, verlässliche Testabdeckung – insbesondere im Angular-Frontend – sowie eine nahtlose Integration in bestehende Entwicklungs- und Betriebswerkzeuge.

VORGEHEN

Zu Beginn analysierte unser Consultant die bestehende Codebasis im Hinblick auf KI-Tauglichkeit, Architekturvorgaben und Teststruktur. Darauf aufbauend wurde gemeinsam mit einem Coding Agent ein konkreter Umsetzungsplan erarbeitet. In einem nächsten Schritt entstanden projekt- und repospezifische Rules und Skills, mit denen Programmierrichtlinien, Testvorgaben und Architekturprinzipien explizit für die Agenten abgebildet wurden.

Erste KI-Ergebnisse wurden systematisch gesichtet, verfeinert und an die definierten Regeln angepasst. Parallel etablierte der Consultant einen klaren Workflow aus Analyse („read“), Implementierung mittels TDD und fortlaufender Dokumentation in Lessons-Learned-Artefakten. Auf dieser Basis wurde die Agentic-Coding-Landschaft iterativ weiterentwickelt und um spezifische Agenten für Kiro ergänzt. Die Architektur und das Systemverhalten der Agenten wurden gezielt gesteuert, um eine konsistente Umsetzung über Frontend und Backend hinweg sicherzustellen. Die enge Zusammenarbeit mit dem Umsetzungsteam stellte sicher, dass Erfahrungen kontinuierlich in Konfiguration und Regeln zurückflossen.

ERGEBNIS

Zum Projektende war Agentic Coding in dem Projekt fester Bestandteil des Entwicklungsalltags. Gut beschriebene JIRA-User-Stories können seitdem mit einem Prompt umgesetzt werden. Der erzeugte Code entspricht den definierten Coding-Conventions und enthält die erforderlichen Tests. Insbesondere im Angular-Frontend führte dies zu spürbar höherer Testqualität und besserer Standardisierung. Architekturvorgaben werden konsistent eingehalten, da sie explizit in Rules und Agentenlogik verankert sind. Zusätzlich unterstützt die KI das Team bei der Analyse von Logs und Produktionsauffälligkeiten, wodurch Fehler schneller eingegrenzt werden können. Die tägliche Nutzung und der kontinuierliche Austausch im Team belegen die Akzeptanz des Ansatzes. Agentic Coding wird dabei bewusst als fortlaufender Lern- und Verbesserungsprozess verstanden.

VERSICHERUNG

Unser Kunde verfolgte das Ziel, große Teile der Softwareentwicklung durch KI-gestützte Agenten zu automatisieren. Wir unterstützten beim Aufbau einer Agentic-Coding-Landschaft, die User Stories gemäß Architektur- und Qualitätsvorgaben eigenständig bis in das Repository umsetzt und zugleich Analyse und Fehlersuche im Betrieb ermöglicht.

TECHNOLOGIEN & METHODEN

- **Claude Code**
- **Kiro (Agent Runtime)**
- **Agentic Coding, TDD**
- **Context7**
- **Dynatrace**
- **Jenkins**
- **Skills & Rules**
- **Atlassian Jira**
- **Atlassian Confluence**
- **Git-Repository**
- **Grafana**