KI gestütze App Entwicklung

Spec-Driven & Test-Driven Development

Team

Sofia Osterland
Mobile Medien, HdM Stuttgart

Nick Berger
Mobile Medien, HdM Stuttgart

Motivation / Problemstellung

- KI-App-Entwicklung soll nicht aus „prompten & hoffen“ bestehen.
- Wir brauchen einen gut nachbaubaren Prozess mit klaren Rollen, Tests und Reviews.
- Wichtige Fragen: Qualität, Nachvollziehbarkeit, Datenschutz und Cloud-vs-Lokal.

Umsetzung / Lösung

Spec-driven und test-driven Workflow direkt in VS Code: erst Spec freigeben, dann Code erzeugen, lokal testen, Git-Status prüfen und am Ende menschlich reviewen.

Flutter | VS Code | Git | Claude Code + Codex | Ollama + Continue.dev | Qwen / DeepSeek

Nicht prompten & hoffen - sondern Prozess

Unser Kernprinzip: Jede KI-Ausgabe läuft durch klare Gates.

1

Spec freigeben

kein Code

Anforderungen und Akzeptanzkriterien zuerst.

→

2

Lokal testen

kein Commit

Tests lokal ausführen und Fehler sichtbar machen.

→

3

Git-Status prüfen

kein Push

Diff kontrollieren, Arbeitsstand sauber halten.

→

4

Mensch reviewt

keine Freigabe

Finale Entscheidung bleibt beim Team.

Ziel: ein reproduzierbarer KI-App-Entwicklungsprozess, der Qualität und Nachvollziehbarkeit statt Zufall liefert.

Cloud / Lokale-KI direkt in VS Code

Cloud-KI

Claude Code + Codex

Spec schreiben	**GPT 5.5**
Spec reviewen	**Claude Opus 4.8**
Code schreiben	**Claude Opus 4.8**
Code reviewen	**GPT 5.5**
Tests schreiben	**GPT 5.5**
Tests reviewen	**Claude Opus 4.8**

Lokale-KI

Ollama + Continue.dev

Qwen 14B / 32B

14B baut & testet
32B reviewt den Diff

DeepSeek-R1 32B

Architektur & Warum-Fragen

Bei Interesse kommt uns gerne im Modulbau besuchen!