

# Analyse der KI-Unterstützung (AI Review)

**Projektmitglieder:** Vanessa Schodl, Samara Strasil (Team „Pomodoro“)

Rainer Adelsberger, Janine Schrahböck (Team „Converter“)

**Projekte:** Pomodoro-Timer & Converter, FHW-DI / WS2025/26

## 1. Methoden der Code-Generierung & Input - VIBE-Coding

- **Input-Strategie:** Wir haben KI (ChatGPT) gezielt – mit „vibe-Coding“ für die Erstellung mathematischer Logik-Bausteine und Umrechnungsformeln genutzt.
- **Iterativ-kollaboratives Arbeiten:** Beide Gruppen haben Schritt für Schritt modular gearbeitet, dh., via vibe-Coding einzelne Teile (Module, Klassen, ... ) erstellt und diese dann später zusammengeführt. Bspw wurde die KI mit dem Command / Input „Erstelle eine Python-Funktion für die Umrechnung von Celsius in Fahrenheit mit Fehlerbehandlung für ungültige Datentypen“ belegt. Es ergaben sich dabei teilweise Fehlfunktionen und Ineffizienzen. Die wesentliche Aufgabe im Schaffungsprozess war sohin das Code-Review (eigen, fremd). Dabei wurden nicht alle Fehler manuell beseitigt – die Lösung bestand oft darin, zu erkennen und darauf basierend den zur Lösung führenden Prompt an die KI zu entwickeln.

## 2. Funktionalität und Qualität des KI-Codes

- **Struktur:** Der generierte Code war erstaunlich „gut“ und – mit unserem Kenntnisstand - verständlich, erforderte aber teilweise Nachbearbeitung/en.
- **Korrektheit:** Der von der KI generierte Code war stets grundsätzlich funktional und die Ergebnisse (Eingaben → Ausgaben) stimmten idR auf Anhieb mit dem Design überein. Die modulare Bauweise und die zunächst isolierte Entwicklung von Software-Teilen führte dazu, dass beim Zusammensetzen zum Ganzen teilweise nachgearbeitet werden musste. Teilweise stießen wir aber auch auf (aus unserer Sicht) unerwartete Probleme;
- **(zB.:) Datentypen:** Die KI verwendete meist korrekte Datentypen (wie `float`), jedoch mussten wir manuell sicherstellen, dass Benutzereingaben über `input()` konsequent validiert werden, um Laufzeitfehler zu vermeiden.

### 3. Vergleich: Mensch vs. KI

- **KI-Leistung:** Enorme Geschwindigkeit beim Bereitstellen von Standardformeln und Grundgerüsten für if-else-Menüstrukturen. Mit Hilfe der KI konnten in beiden Projekten tragfähige Grundgerüste geschaffen werden.
- **Menschliche Leistung:** Während die KI oft nur lineare Abläufe vorschlägt, haben wir eine robuste Benutzerführung programmiert bzw. die KI dahingehend gedrillt (→ vibe-Coding). Wir mussten ineffiziente oder redundante Operationen der KI entfernen, um den Code schlanker zu gestalten. Dies war möglich, weil wir auch mit unserem Kenntnisstand bereits grundsätzlich verstehen können, was (in etwa) eine Codezeile machen wird. Darauf basierend bzw. mit diesem Verständnis war uns sodann eine entsprechende Adaptierung nach unseren Design-Vorstellungen möglich.
- **Mensch-Maschinen-Kommunikation:** Die KI produzierte oft eher generische Fehlermeldungen, die eine/n Anwender/In ohne dedizierte Kenntnisse womöglich unverständlich wären. Durch unser diesbezügliches Verständnis konnten wir die generischen Ausgaben der KI in „verständliche Sprache“ bringen.

### 4. Dokumentation und Standards

- **Dokumentationsdefizit:** Der generierte Code enthielt zwar viele einfache Auskommentierungen (#), jedoch fehlten fast vollständig die für Python-Standards wichtigen Docstrings (""""..."""").
- **Wartbarkeit:** Um die Wartbarkeit und Skalierbarkeit zu gewährleisten, haben wir diese Docstrings manuell nach dem PEP 257 Standard ergänzt, damit die Funktionsweise auch für externe Entwickler ohne Lesen des Quellcodes verständlich ist.

### 5. Optimierung

- **Optimierung:** Wir haben sichergestellt, dass keine redundanten Berechnungen durchgeführt werden, was den Code im Vergleich zum ersten KI-Entwurf effizienter macht. Dafür mussten auch manuelle Anpassungen im Code vorgenommen werden.

**Fazit der Entwickler-Teams:** Die KI war ein absolut notwendiges und dementsprechend hilfreiches Werkzeug, um die mathematische wie auch syntaktische Basis des Projekts schnell aufzubauen. Die eigentliche Software-Engineering-Leistung bestand für uns darin, den Code **robust gegenüber fehlerhaften Benutzereingaben** zu machen und die **Dokumentationsstandards** (Docstrings) einzuhalten, die von der KI vernachlässigt wurden.