

AI-Augmented Software Development Beyond Code Generation

Towards Maintainable, Controllable and Sustainable Software Engineering

Starting Position

Artificial Intelligence – especially large language models – is now widely used in software development. Developers increasingly rely on AI-based tools for code generation, refactoring, testing, and documentation. Early evidence suggests that these tools significantly increase development speed and release frequency.

However, recent industry reports (e.g., from Google/DORA) indicate a growing tension: while release rates and development throughput increase, system stability and reliability often decline. This points to a fundamental challenge – AI can accelerate software production, but not necessarily its quality. Several issues contribute to this problem:

- **Lack of transparency:** AI-generated code often lacks clear rationale, making it harder to understand, review, and maintain
- **Reduced control:** Developers may rely on generated solutions without fully understanding their implications
- **Quality assurance challenges:** Traditional testing and review practices are not always adapted to AI-generated artefacts
- **Erosion of architectural consistency:** AI-generated code may introduce inconsistencies across a system over time
- **Short-term optimization vs. long-term sustainability:** Increased speed can come at the cost of maintainability and evolvability

These challenges raise fundamental questions about how software engineering must evolve in the age of AI. Rather than focusing solely on productivity gains, there is a need to understand how AI can be integrated into development processes in a way that ensures quality, control, and long-term sustainability.

Within this research project, we aim to explore what “serious AI-augmented software development” means – i.e., how AI can be used as a tool to support professional software engineering practices without compromising system quality.

Goals / Approach

The goal of this master project is to investigate and develop approaches for AI-augmented software development that go beyond code generation, focusing on control, quality, and sustainability.

- **Maintainability and evolution:** Analyse how AI-generated systems can be maintained and evolved over time.
Example: Comparing AI-generated and manually written code in terms of structure and changeability.
- **Quality assurance:** Investigate methods to assess and improve the quality of AI-generated software.
Example: A validation pipeline combining tests, static analysis, and review support.
- **AI-augmented development workflows:** Design and implement workflows where developers specify requirements and constraints, AI generates artefacts, and validation mechanisms ensure correctness.
Example: AI generates a feature and tests, followed by automated validation (e.g., coverage, static checks).
- **Control and traceability:** Develop approaches to keep developers in control of the system, by making AI-generated artefacts transparent and traceable and decision support.
Example: Linking requirements and prompts to generated code.

Evaluate the approaches in realistic scenarios and derive principles for effective and sustainable AI-augmented software engineering.

Tasks for the Master student

The interested student can select from the wide range of challenges listed above according to their interests. Develop concepts, and prototypes, implement solutions, and validate them.

Study type: [X] Fulltime
[X] Part time 50% employed as research assistant

Project organisation: Single and/or as part in a project team

Project finance: internal

Work place: Windisch

Advisor: Prof. Martin Kropp