From Assistive AI to Agentic Engineering: AI-Supported Software Development Across the V-Model

Dragan Milchevski, Vehicle Motion, Robert Bosch GmbH (Dragan.Milchevski@de.bosch.com)

Abstract

AI-based methods promise substantial productivity gains in software development, but their safe and verifiable use in safety-critical software development remains unresolved. To address this challenge, we introduce an agentic AI system for software development in the automotive domain that ensures verifiability by providing end-to-end traceability across the V-Model. Our system assists with a wide range of activities, from requirements analysis and architecture design, software implementation, test derivation, and quality review.

We distinguish five levels of AI-supported systems engineering, adapted from human-automation and AI-agent autonomy taxonomies and defined by delegated engineering authority. Level 1, Assistive AI, supports engineers through retrieval, summarisation, critique and alternative generation without authority to modify artefacts. Level 2, Augmented Decision Support, introduces task-specific agents that create structured, citable proposals for individual engineering activities, with engineers responsible for acceptance. Level 3, Gated Workflow Agency, allows agents to maintain state across linked V-Model artefacts, invoke tools or specialised agents, and propagate proposed changes under explicit human approval gates. Our current work is positioned between Levels 2 and 3. Moving to Level 4 would require conditionally autonomous engineering loops that execute bounded workflows while escalating uncertainty, safety impact, process deviations or insufficient evidence. Level 5 is concerned with full lifecycle of autonomy, which remains a research horizon; clarifying the assurance, qualification, accountability and safety-case evidence is required before its realization.
In the Bosch case, customer requirements provide the entry point into a connected engineering workflow rather than the sole object of automation. Starting from incoming specifications and platform baselines, the system analyses coverage and impact, identifies affected functions and architectural elements, and coordinates specialised agents for downstream tasks. Depending on the change context, these agents may propose derived system, software or component-level artefacts; support architecture for SW co-design decisions; assist software component creation and code generation; and prepare unit, integration and system tests.

Our system combines multi-agent orchestration with a property-graph representation of the engineering corpus, enabling agents to navigate dependencies across requirements, functions, architectures, design models, code, tests, and safety-relevant evidence. Every AI-generated finding or proposal is linked to a reproducible evidence trail containing the artefact fragments, engineering rules, prompts, model outputs, tool calls and graph relations used to produce it. This makes agentic task execution reviewable and provides the basis for moving beyond Level 3 without losing traceability, tool confidence or safety-case relevance. Finally, we compare our approach with traditional manual workflows, showing where AI already reduces effort, where it introduces new verification and governance risks, and which open research questions must be resolved before higher autonomy levels can be justified.