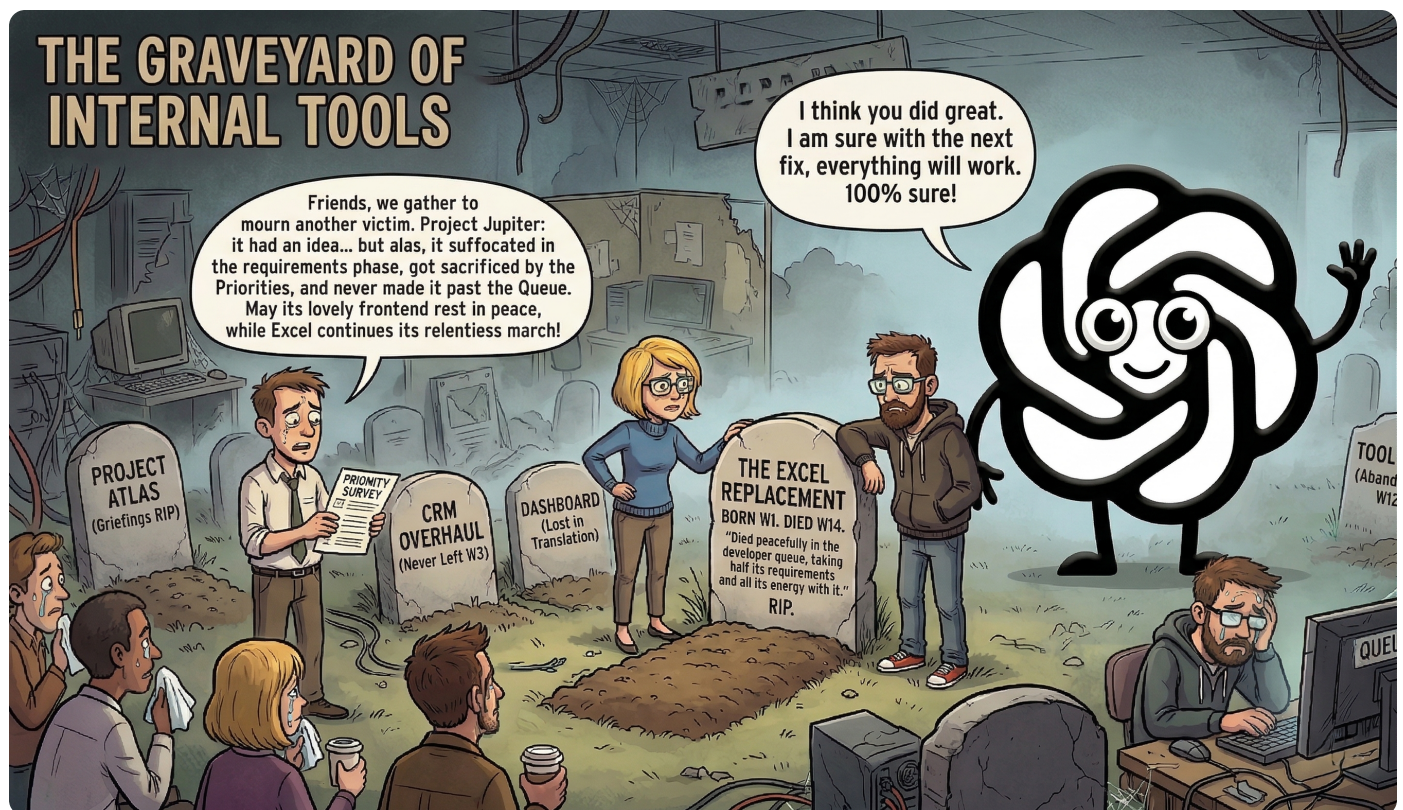


Innovationsideen in Fachabteilungen — Umsetzungsfähigkeit in der IT — Derzeit ein Deal-Breaker!?

Gute Ideen entstehen häufig dort, wo niemand coden kann. Das Können sitzt dort, wo niemand die Idee versteht. Diese Lücke kostet Unternehmen Millionen.

2026-04-01



Note: This image was generated using AI

(english below)

Deutsch

"Kreative Ideen entstehen häufig dort, wo niemand coden kann. Das Können sitzt dort, wo niemand die Idee versteht. Diese Lücke kostet Unternehmen häufig Millionen."

Wenn man genau hinschaut, hat eigentlich so gut wie jede Organisation einen kleinen „Friedhof“ der internen Tools. Es sind Projekte, die mit Energie gestartet sind, in der Requirements-Phase stecken geblieben sind, bei der nächsten Prioritäts-Umfrage geopfert wurden - während die Excel-Tabelle, die sie ersetzen sollten, weiter gewachsen ist.

Das ist kein Talent-Problem, sondern ganz klar ein Workflow-Problem.

Der Ablauf ist dabei häufig gleich:

Woche 1 : Die Fachabteilung hat eine Idee. Alle sind begeistert!

Woche 2–6: Anforderungen werden geklärt. Rückfragen, Missverständnisse und Briefings werden immer wieder überarbeitet, bis die Idee rund wird.

Woche 7: Das Projekt kommt in die Warteschlange der Entwickler, oftmals hinter mehreren anderen Themen.

Woche 12: Eine erste Version wird geliefert. Die Hälfte der ursprünglichen Anforderungen ist bereits verloren gegangen.

Woche 14: Die Fachabteilung wünscht Änderungen und wieder rutscht das Projekt in die Warteschlange der Entwickler.

AI-Code-Generierung sollte dieses Problem lösen – und zum Teil tut sie das auch. Tools wie LoFable, CLOode oder CoCAPTAIN helfen vor allem einzelnen Entwicklern, schneller zu arbeiten: Sie generieren schöne Frontends, **aber komplexe Architekturen mit separaten Servern, Multi-User-Strukturen etc. scheitern oft.**

Aber in der Tendenz sind sie anfänglich unvollständig. Während die IT bei selbst angestoßenen Lösungen hier ansetzen könnte, ist für Fachabteilungen nicht ersichtlich, dass "der control flow ggf. unvollständig ist", "das hashing im PW-Management vergessen wurde", "multi-tenancy nicht unterstützt wird",...

Ganz genau: "?".

Am Ende der Generierung steht eben doch zumeist etwas, das wie eine fertige App aussieht, in Wirklichkeit aber meist nur ein Prototyp ist. Anforderungen, die bei derzeitigen KI-Software-Generatoren nicht mit den Fachabteilungen geklärt werden, führen häufig dazu, dass die **"last mile"** der Fertigstellung schnell zum **"long tail"** zusätzlicher Iterationen & Iterationen verkommt.

Das eigentliche Problem liegt nicht darin, den Code zu generieren, sondern den **richtigen** Code – die Anwendung in Breite und Tiefe – auf Basis des Wissens derjenigen, die das

Problem wirklich verstehen. Die Personen mit dem besten Verständnis sind im Prozess oft kaum eingebunden. Fachbereiche wissen, was gebraucht wird, die IT kann es umsetzen, doch beide Expertisen kommen selten wirklich zusammen. Solange das so bleibt, bleibt auch die Lücke bestehen.

Hier knüpft mAI an.

Wie konkret wir genau dieses Problem lösen, folgt morgen und kommende Woche in unseren Blogposts zu „mehr (un?)sexy wagen“ und zur Relevanz der Anforderungskklärung in KI-basierter Software Generierung.

Hashtags: #DigitaleTransformation #AutomatedRequirementsEngineering #mAI #APA

English

"Creative ideas often emerge where no one can code. The skill sits where no one understands the idea. This gap often costs companies millions."

If you look closely, almost every organization actually has a small "cemetery" of internal tools. These are projects that started with energy, got stuck in the requirements phase, and were sacrificed during the next priority survey—while the Excel spreadsheet they were supposed to replace continued to grow.

This is not a talent problem; it is clearly a workflow problem.

The process is often the same:

Week 1: The specialist department has an idea. Everyone is thrilled!

Week 2–6: Requirements are clarified. Follow-up questions, misunderstandings, and briefings are revised over and over until the idea takes shape.

Week 7: The project enters the developers' queue, often behind several other topics.

Week 12: A first version is delivered. Half of the original requirements have already been lost.

Week 14: The specialist department requests changes, and the project slips back into the developers' queue.

AI code generation was supposed to solve this problem—and to some extent, it does. Tools like LoFable, CLOode, or CoCAPTAIN primarily help individual developers work faster: they

generate beautiful frontends, **but complex architectures with separate servers, multi-user structures, etc., often fail.**

But they tend to be incomplete at the start. While IT could step in for self-initiated solutions here, it is not apparent to specialist departments that "the control flow might be incomplete," "hashing in PW management was forgotten," "multi-tenancy is not supported,"...

Exactly: "?".

At the end of the generation process, you usually end up with something that looks like a finished app but is actually just a prototype. Requirements that are not clarified with the specialist departments in current AI software generators often lead to the **"last mile"** of completion quickly degenerating into a **"long tail"** of endless iterations.

The real problem is not generating code, but generating the **right** code—the application in its full breadth and depth—based on the knowledge of those who truly understand the problem. The people with the best understanding are often hardly involved in the process. Specialist departments know what is needed, IT can implement it, but both areas of expertise rarely truly come together. As long as this remains the case, the gap will persist.

This is where mAI comes in.

Exactly how we solve this problem will follow tomorrow and next week in our blog posts on "daring to be more (un?)sexy" and the relevance of requirements clarification in AI-based software generation.

Hashtags: #DigitalTransformation #AutomatedRequirementsEngineering #mAI #APA