

Kill the Vibe?

Architektur im KI-Zeitalter

Stefan Toth

Jax, 05/26





Stefan Toth

CEO, Berater für Agilität
Softwarearchitektur



Stefan.Toth@embarc.de

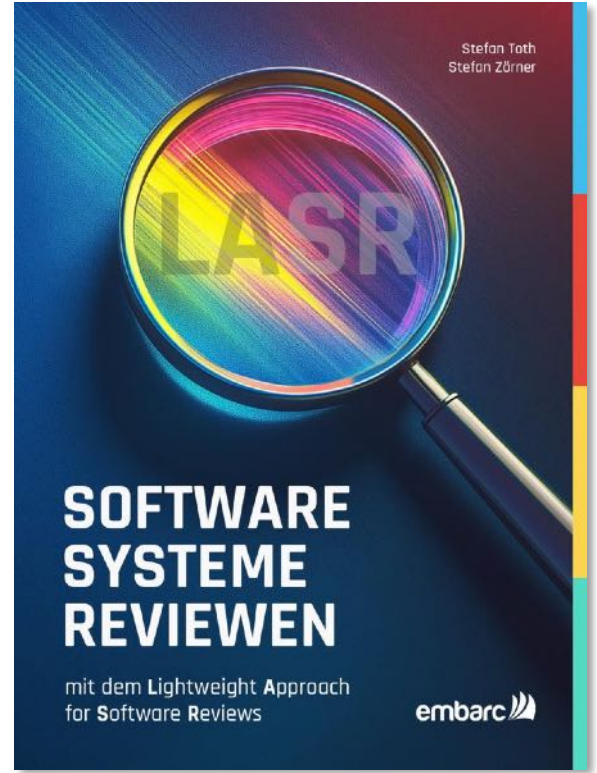


[linkedin.com/in/sto-embarc](https://www.linkedin.com/in/sto-embarc)



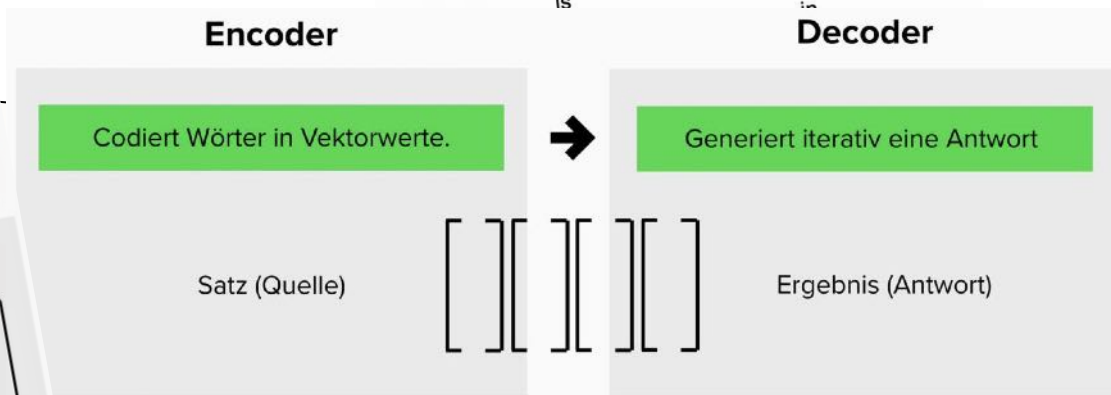
www.embarc.de



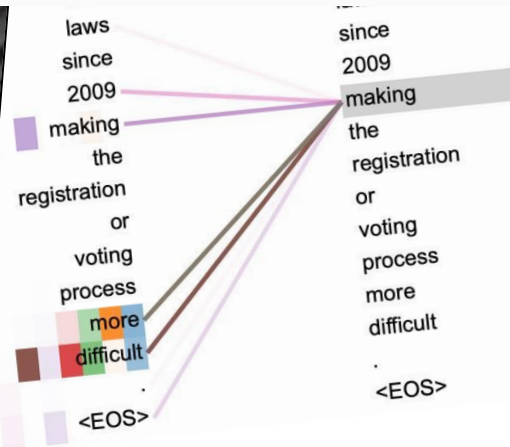




Transformer-Modelle / LLMs



esezeit: 7 Minuten



Decoder

Generiert iterativ ("auto regressiv") eine Antwort



Startpunkt /
bisheriges Ergebnis

"Masked Self-attention"

Versucht Sätze sinnvoll weiter zu
spinnen, indem der bisherige Satz
(die Wörter links davon) als Kontext
dienen (rechts ist "maskiert").

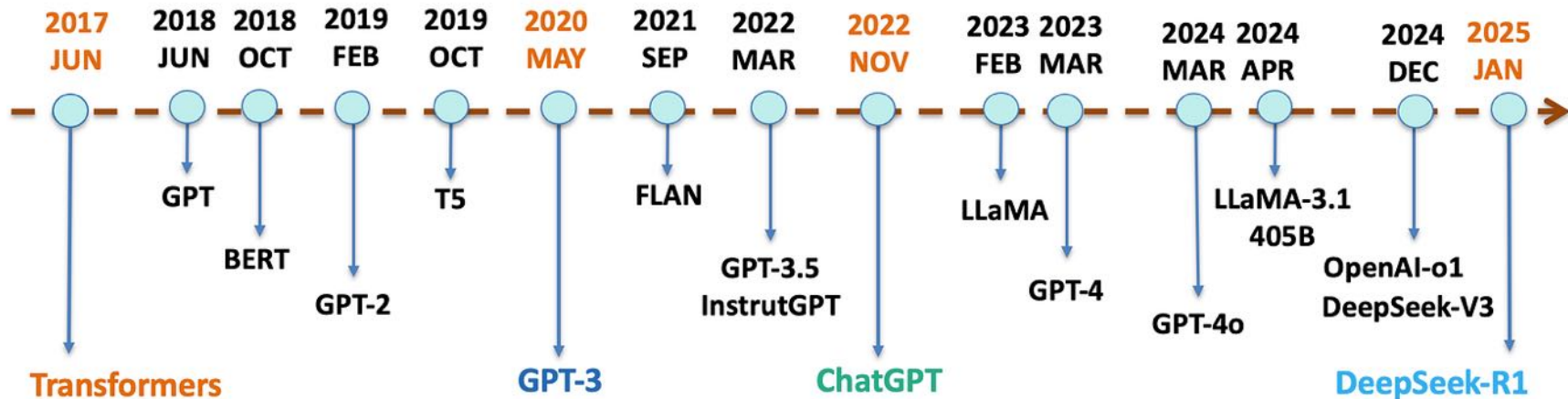
- Eine
- Eine sinnvoll
- Eine sinnvoll formulierte
- Eine sinnvoll formulierte Antwort



Blogpost

Die Welt explodiert

A Brief History of LLMs



Kill the Vibe?

Architektur im KI-Zeitalter

Stefan Toth

Jax, 05/26



Worüber ich heute spreche...

Architektur-Methodik



Systeme mit Gen-AI-Mitteln designen und weiterentwickeln

Architektur-Lösungen



Architektur-Ideen für Softwaresysteme mit KI-Anteilen

AI & Architektur

- relativ einfach umzusetzen
- benötigt Human-in-the-loop und/oder spezifisch entwickelte Lösungen
- ist schwierig oder fehleranfällig, Unterstützung von Entwicklerinnen durch AI-Lösungen ist möglich

Automatisierte Unterstützung bei Architekturdokumentation

Domänenschnitt und Gliederung entwerfen / reflektieren

Inkonsistenzen im Code entdecken

Abgleich von vorhandener Dokumentation und Code

Architektur-Reviews automatisiert durchführen

Root-Cause Analyse bei Problemen (Logs, Monitoring, ...)

Dokumentationsqualität prüfen: Lücken, Inkonsistenzen, ...

Technologie-Evaluation Make/Buy bzw. Wiederverwendung

Passenden Architekturstil finden und ausprägen

Änderungsaufwand für größere Umstellungen schätzen

Legacy- oder Fremdlösungen besser verständlich machen

Impact von Änderungen auf Qualitätsziele abschätzen

Technologie/Framework Map nach Verwendung, letztes Update, Aktivität, ...

Bekannte Bugs & Security Issues in genutzten Technologien sichtbar machen

Architecture Decision Records dokumentieren

Unterstützung bei der Identifikation und Ausarbeitung von Qualitätszielen

Kontextspezifischen Architekturprinzipien entwickeln

Lösungsoptionen für spezifische Probleme suchen



Vibe

[vaɪb]

That intangible something that tells you whether someone's about to offer you cookies or aggressively mansplain quantum physics.



Vibe



Good!

[vaɪb]

That intangible something that tells you whether someone's about to **offer you cookies** or aggressively mansplain quantum physics.



Vibe-Coding

[vaɪb koʊ.dɪŋ]

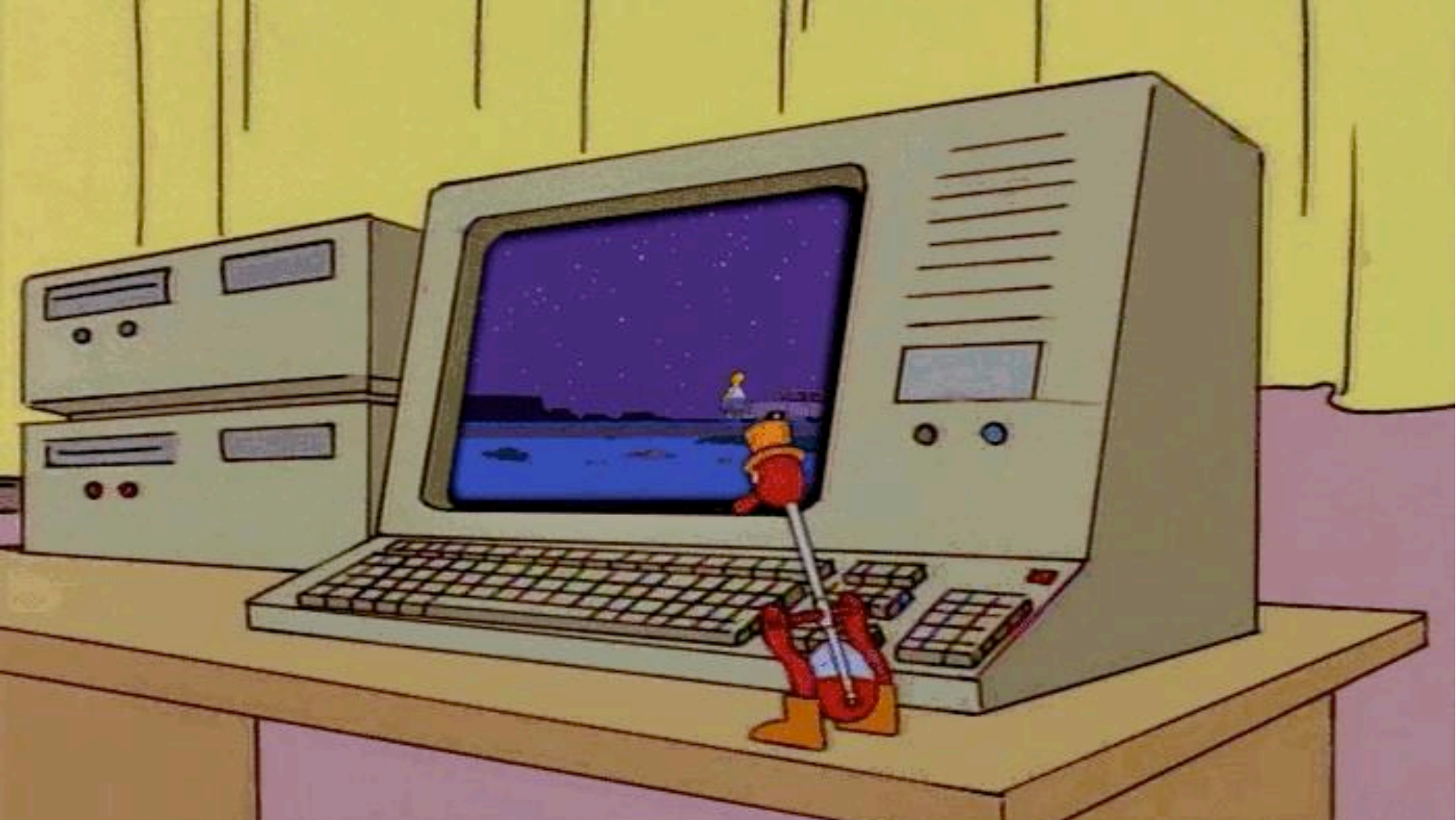
In seiner explorativsten Form vertraut der Nutzer vollständig darauf, dass der KI-Output wie beabsichtigt funktioniert. Wie *Karpathy* es formulierte, ist das so, als würde man „vergessen, dass der Code überhaupt existiert“.



Vibe-Coding

[vaɪb koʊ.dɪŋ]

In seiner explorativsten Form vertraut der Nutzer vollständig darauf, dass der KI-Output wie beabsichtigt funktioniert. Wie *Karpathy* es formulierte, ist das so, als würde man „vergessen, dass der Code überhaupt existiert“.



F*** yeah!

Great news! The scraping is working perfectly now:

1. **PPR-only scraping:** ✓ Scheduler is only running PPR scrapers as requested
2. **Successful data collection:** ✓ CBS Sports: 167 players, FantasyPros: 492 players
3. **Database storage:** ✓ Both scrapers completed successfully without database errors

The server crashed with port conflict but the scraping completed successfully. Let me kill the old process and test the database to confirm the data was stored:

```
Bash(lsof -ti:3001 | xargs kill -9)
└─ (No content)
```

- Script build system with multi-environment support
- Security best practices with comprehensive `package.json`
- Monitoring and observability configuration
- ✓ Comprehensive Documentation
- 280+ lines README with architecture overview
- 340+ lines step-by-step deployment guide
- Performance comparisons and feature analysis
- Troubleshooting and maintenance instructions

• 🏆 HIVE MIND COLLECTIVE INTELLIGENCE - MISSION ACCOMPLISHED! 🏆

🏆 EXTRAORDINARY SUCCESS: COMPLETE WEB SOLUTION DELIVERED

Your BGG Checker has been fully transformed into a modern, scalable web application with feature parity and enhanced performance.

DELETED

↳ local script into a modern, scalable web

'api/check-availability

thon)



Andrej Karpathy ✓
@karpathy

There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists.

Vibe-Coding

[vaɪb kɔʊ.dɪŋ]

In seiner explorativsten Form vertraut der Nutzer vollständig darauf, dass der KI-Output wie beabsichtigt funktioniert. **Wie Karpathy es formulierte, ist das so, als würde man „vergessen, dass der Code überhaupt existiert“.**

A young man and woman are sitting on a concrete ledge in the foreground, laughing heartily and holding white coffee cups. They are dressed in winter jackets. The background is a scene of total destruction, with a large pile of rubble, twisted metal, and a partially destroyed red brick building. A yellow excavator is visible in the background. The overall atmosphere is one of resilience and joy in the face of adversity.

Bad Vibe

Good Vibe



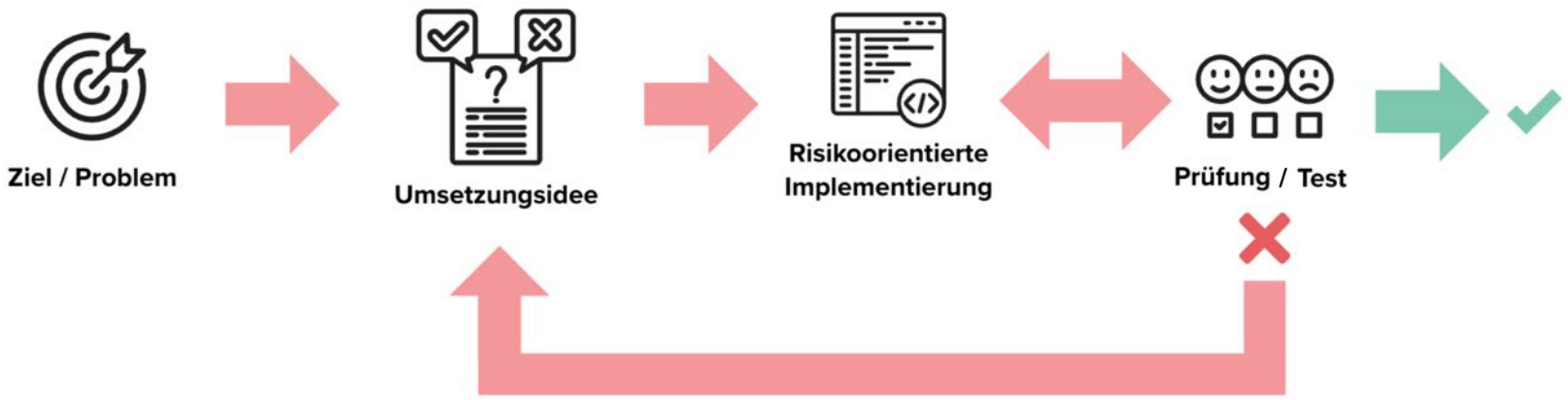
Software
Architecture
Rules!

R.I.P.
Vibe
Coding



Wie sorgen wir für ‚good
vibes‘, ohne ignorant oder
oberflächlich zu sein?

Zielorientierung, Flow, ausreichend Qualität...



AI Entwicklungsansätze



Idee



Tools

Conversational Programming

Softwareentwicklung im iterativen Dialog mit einem LLM

Chat interfaces (ChatGPT, Claude, IDE plugins), copilots mit chat modes.

Spec-Driven Development (SDD)

One-Shot Implementierung auf Basis von Kontext und Spezifikation in natürlicher Sprache

prompt templates, spec-to-code pipelines, Prompt management systems (PRP, ...)

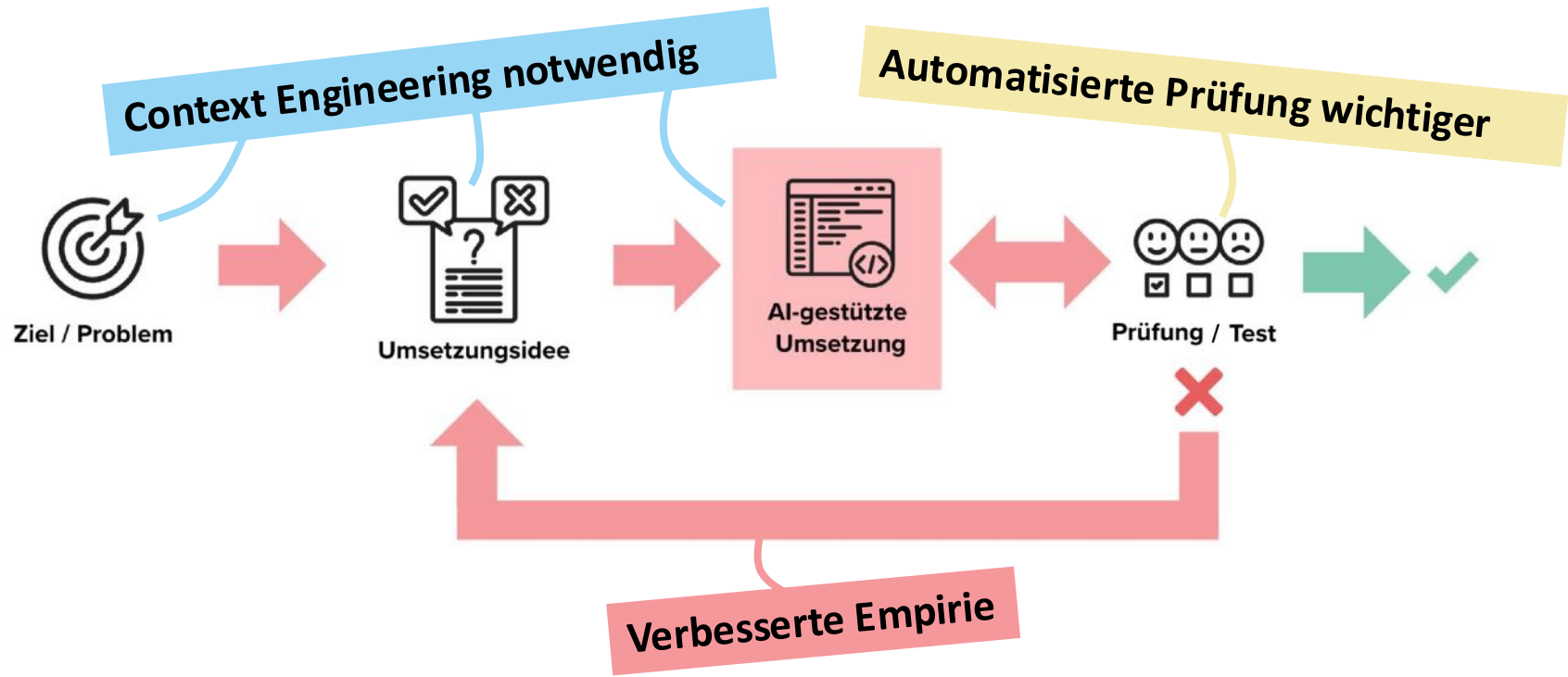
Agentic Development

Umsetzung durch einen oder mehrere autonome Agenten, die planen, reflektieren und umsetzen.

Agents, Agentic frameworks (inkl. Swarms), orchestration runtimes (LangChain, ...), Shared Memory Lösungen

CLI Tools (aider, Claude Code, ...), LLM-first IDEs (Windsurf, Cursor, ...), MCPs, Playgrounds/Sandboxes, Evaluation/Test Harnesses (SWE-bench, ...), ...

Generative KI in der Entwicklung





Context Engineering

Überblick und Steuerbarkeit zurückgewinnen



“Mir sind tausend Nazis lieber
als ein Flüchtling!”

Dinge die unser Chef sagt und falsch verstanden werden.



“Mir sind tausend Nazis lieber
als ein Flüchtling!”

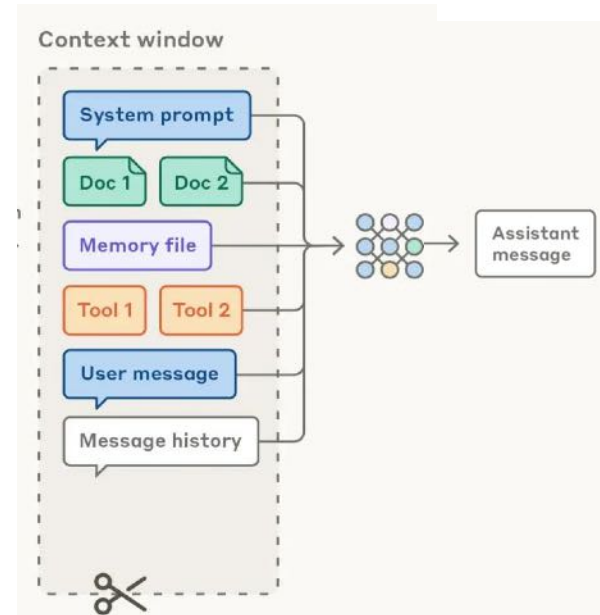
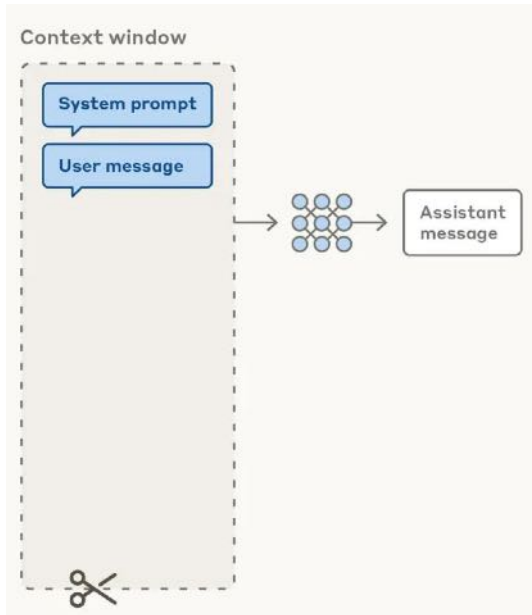
Dinge die unser Chef sagt und falsch verstanden werden.

Hin & Weg
BESTATTUNGEN

Bestattungsvorsorge. Jetzt.
www.HW-BESTATTUNGEN.de

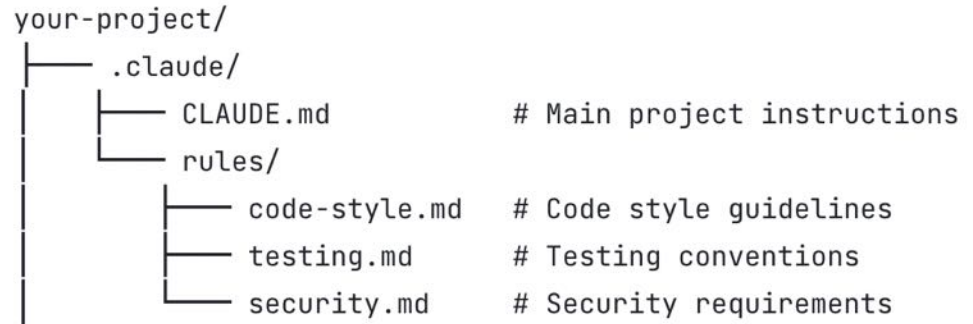
Was ist im Kontext?

Context engineering is the discipline of building dynamic systems that supply an LLM with everything it needs to accomplish a task.



Context Engineering Optionen

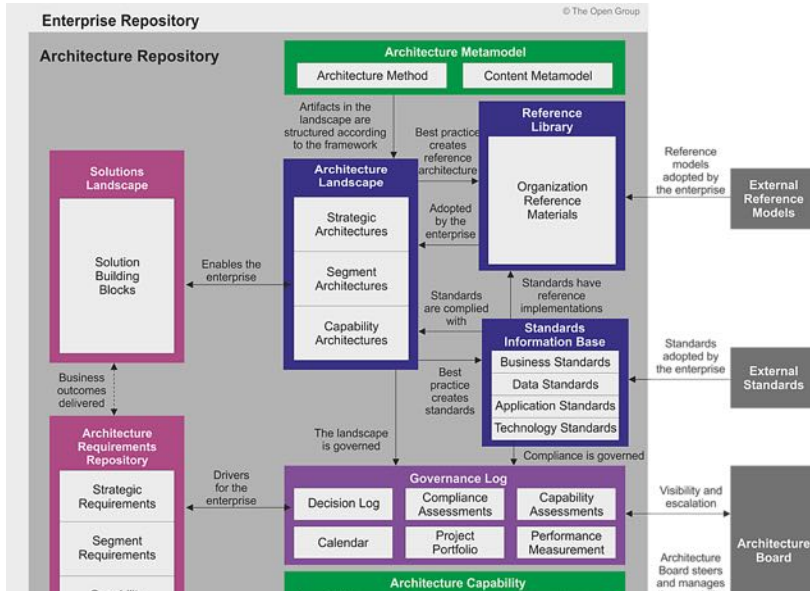
- Claude.md / Agents.md
- Regeln (z.b. als .md)
- System Prompt
- Tools (Fähigkeiten)
- MCP-Server
- Sub-Agenten / Modes (eigener Kontext)
- Skills
- Hooks
- Templates
- ...



```
/context  
├── Context Usage  
│   └── claude-sonnet-4-20250514 • 116k/200k tokens (58%)  
│       ├── System prompt: 2.8k tokens (1.4%)  
│       ├── System tools: 11.6k tokens (5.8%)  
│       ├── MCP tools: 17.0k tokens (8.5%)  
│       ├── Memory files: 178 tokens (0.1%)  
│       ├── Messages: 84.0k tokens (42.0%)  
│       └── Free space: 84.4k (42.2%)  
├── MCP tools • /mcp  
│   └── mcp_applescript_execute_applescript_execute (applescript_execute)588 tokens  
└── mcp_filesystem_read_file (filesystem): 475 tokens  
    └── mcp_filesystem_read_text_file (filesystem): 556 tokens
```

Architektur im Kontext

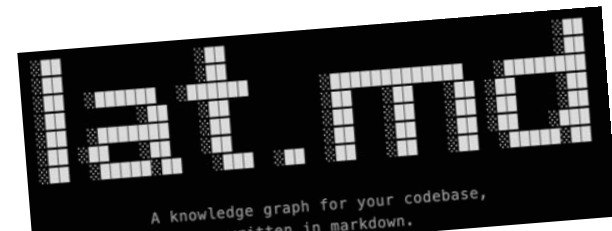
- Domänenkonzepte erklären
- Architekturprinzipien hinterlegen (Agents.md, lat.md, Regeln, System Prompt oder RAG)



```

✓  system-design
   M↓ domain-concepts.md
   M↓ system-overview.md
  
```

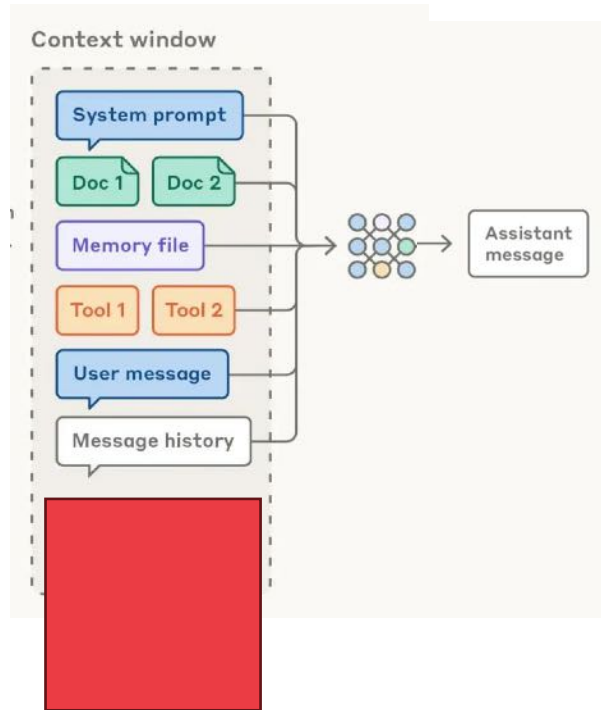
Hintergrund auch bei Nick Tune:
<https://www.oreilly.com/radar/reverse-engineering-your-software-architecture-with-claude-code-to-help-claude-code/>



Architektur im Kontext

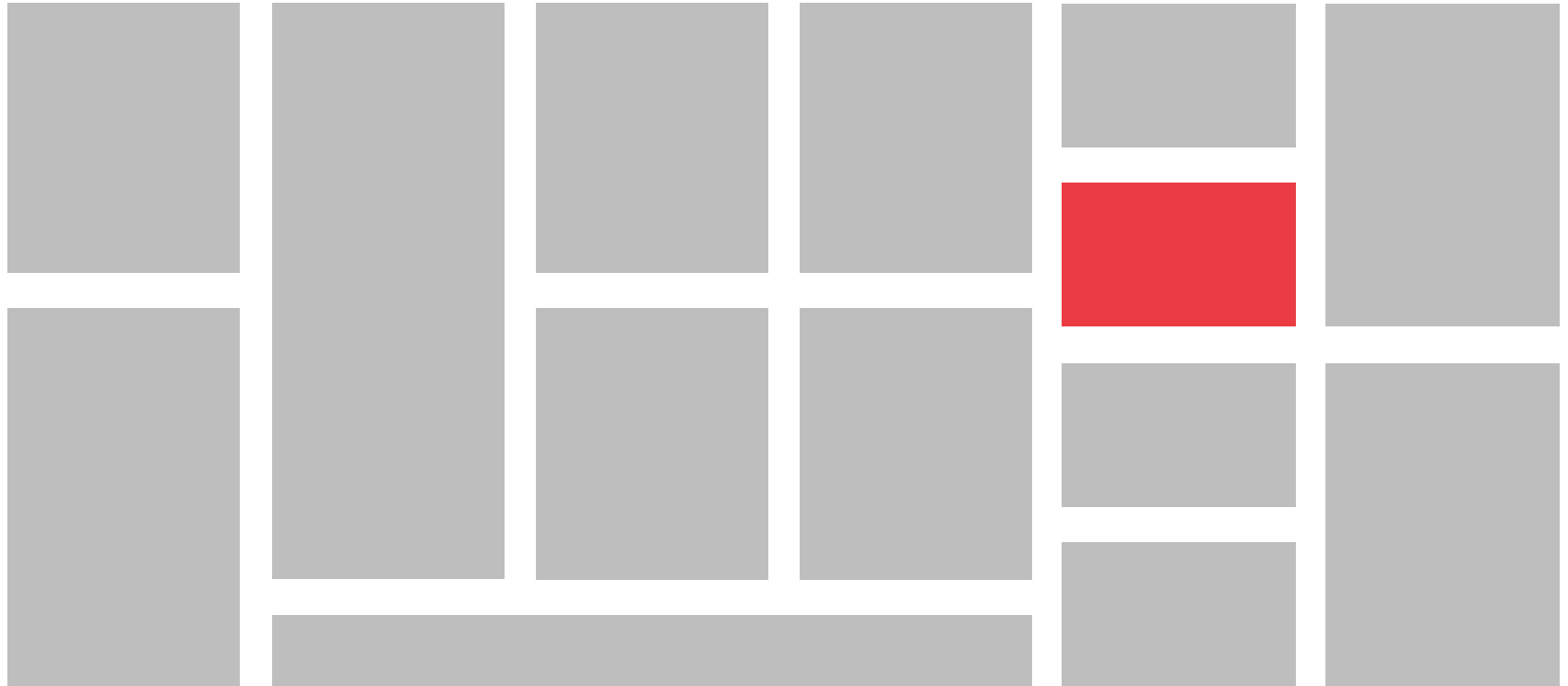
- **Domänenkonzepte** erklären
- **Architekturprinzipien** hinterlegen (Agents.md, lat.md, Regeln, System Prompt oder RAG)
- **Struktur und Abhängigkeiten** definieren (Repos & Rules)
- **Ist-Struktur** der Lösung **zugreifbar** machen: Dependency Graphs oder Repository Planning Graphs (RPG)
- **Spezifische Konventionen** für System-Teile (z.B. Mobile UI) hinterlegen (Skills)
- ...

Context Overflow vermeiden

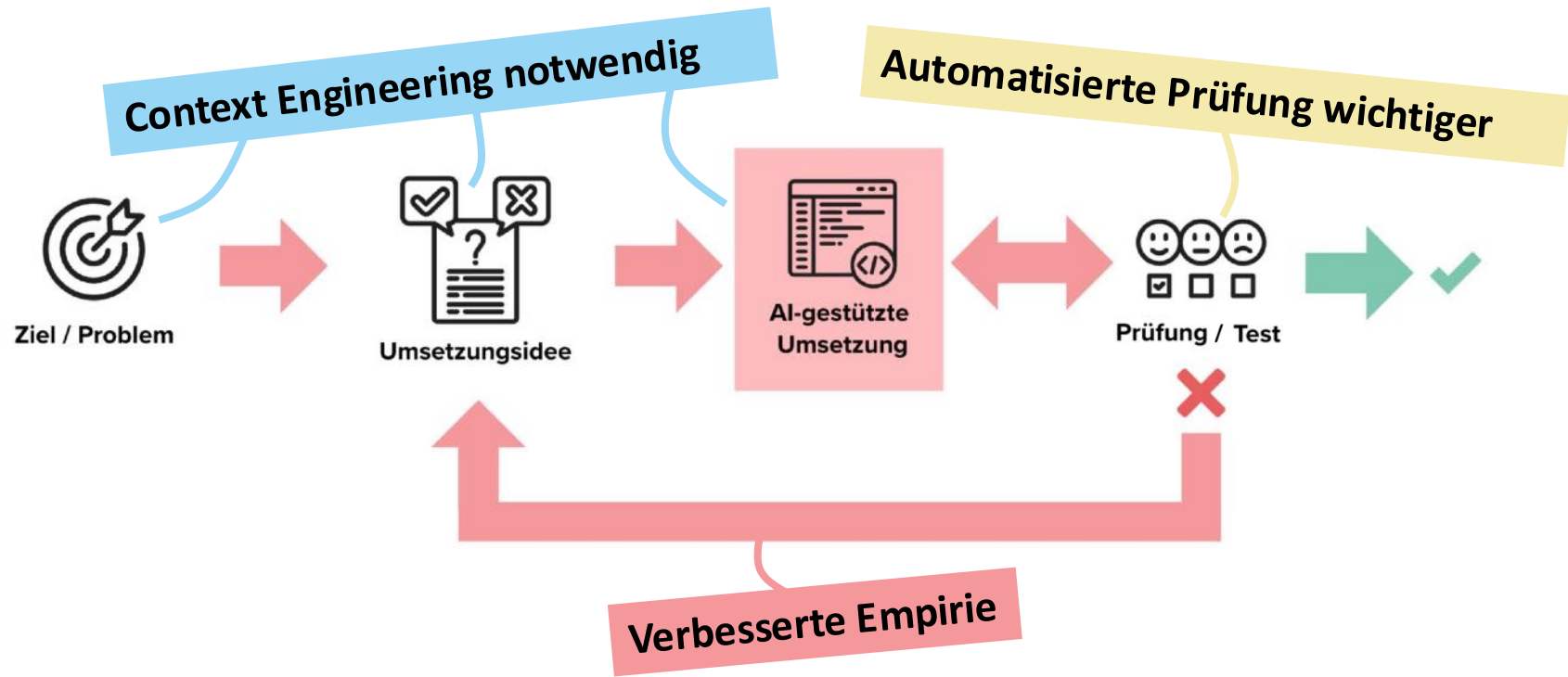


- **Kontext** nach und nach **aufbauen** (statt initial überladen)
- **Sub-Agenten** / Modes nutzen
- **Architekturinformation destillieren** (statt RAG auf alles)
- **Code-Menge reduzieren**
- **Systemteile isolieren**
- Kontext **komprimieren**
- ...

Fehler / Probleme scopen



Generative KI in der Entwicklung





Automatisierte Prüfung

Verzahnung von Architektur und Entwicklung

AI Agents sind mächtig, aber limitiert

```
def test_evidence_collection(self):  
    """Test collecting evidence for validation decisions."""  
    validation = {  
        "stressor_id": "S1",  
        "stressor": True,
```

After thoroughly analyzing the Python test files in /tests/unit, I must deliver a harsh verdict: These tests are completely fake and do not test any actual implementation.

The Fundamental Problem

Language Mismatch:

- Implementation: TypeScript/Node.js (.ts files in /src)
- Tests: Python (.py files in /tests/unit)

This is absurd. You cannot test TypeScript code with Python unit tests.

```
        {  
            "type": "config",  
            "location": "config/database.yml",  
            "snippet": "replicas: [db-replica-1, db-replica-2]"  
        }  
    ]  
}
```

```
assert len(validation["evidence"]) > 0  
assert all("type" in e and "location" in e for e in validation["evidence"])
```

Test / Review Agents + Human in the Loop

The screenshot shows a code editor interface with the following components:

- File Explorer (Left):** Shows a project structure with folders like 'analysis', 'architecture', 'consensus', 'core', 'data', 'development', 'devops', 'documentation', 'github', 'hive-mind', 'optimization', 'sparc', 'specialized', 'swarm', 'templates', and 'testing'. The 'core' folder is expanded, showing files like 'coder.md', 'planner.md', 'researcher.md', 'reviewer.md' (selected), and 'tester.md'.
- Code Editor (Center):** Displays the content of 'reviewer.md'. It starts with a table of contents and then a code snippet for a 'Code Review Agent'.
- Preview Window (Right):** Shows a rendered version of the markdown content.

Code Editor Content:

```
.claude > agents > core > reviewer.md > ...
23 # Code Review Agent
35 ## Review Process
90 ### 3. Performance Review

123
124 ### 4. Code Quality Review
125
126 ```typescript
127 // QUALITY METRICS:
128 ✓ SOLID principles
129 ✓ DRY (Don't Repeat Yourself)
130 ✓ KISS (Keep It Simple)
131 ✓ Consistent naming
132 ✓ Proper abstractions
133
134 // EXAMPLE IMPROVEMENTS:
135
136 // ✗ Violation of Single Responsibility
137 class User {
138   saveToDatabase() { }
139   sendEmail() { }
140   validatePassword() { }
141   generateReport() { }
142 }
143
144 // ✓ BETTER DESIGN:
145 class User { }
146 class UserRepository { saveUser() { } }
147 class EmailService { sendUserEmail() { } }
148 class UserValidator { validatePassword() { } }
149 class ReportGenerator { generateUserReport() { } }
150
151 // ✗ Code duplication
152 function calculateUserDiscount(user) { ... }
153 function calculateProductDiscount(product) { ... }
```

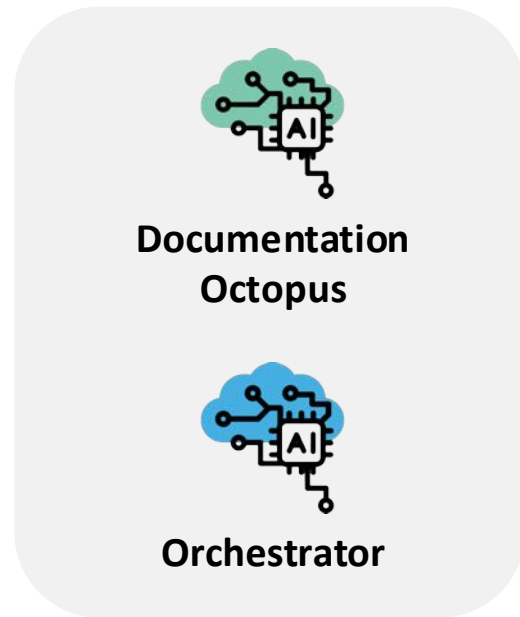
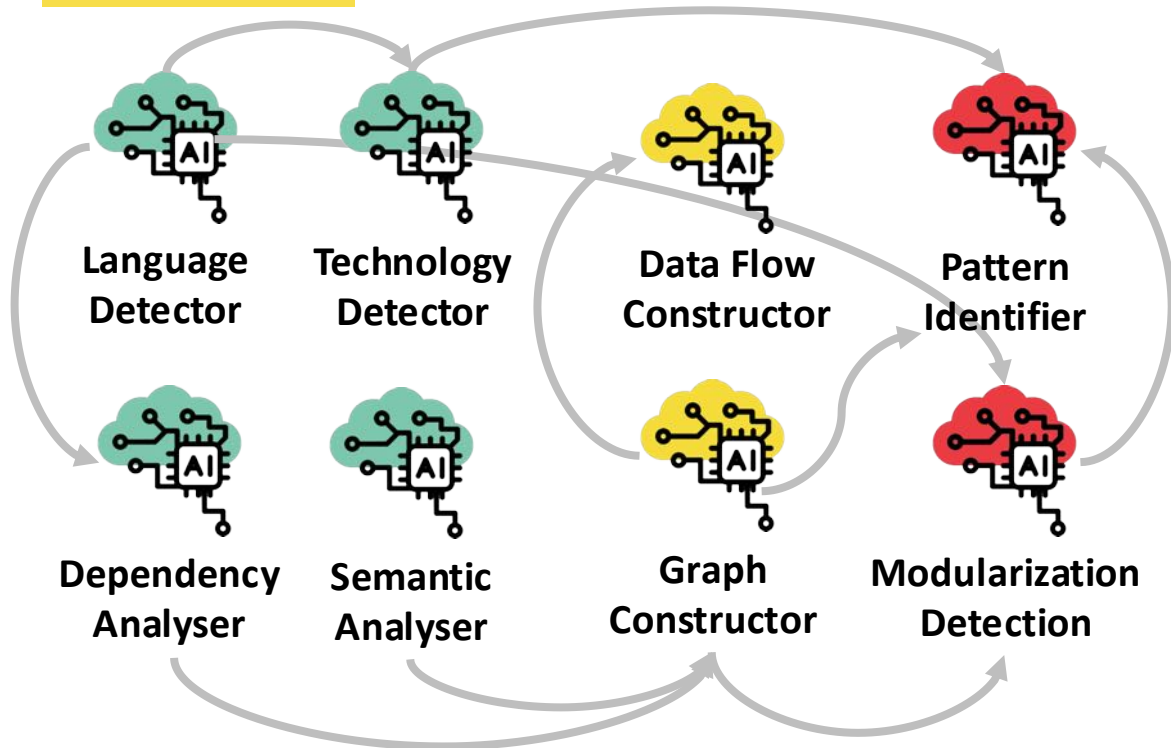
Design Feedback-Loop



The screenshot displays the Sentrux Pro v0.5.3 interface. On the left, a sidebar lists various quality metrics. The main area shows a grid of charts for different metrics. On the right, a detailed view for the 'Quality' metric is shown, including a progress bar and a table of values.

Metric	Value	Target
Quality	2627	-
modularity	Q=1.00	10000
acyclicity	0 cycles	10000
depth	0 max	10000
equality	G=0.88	1250
redundancy	10000	0
edges	0/0	files 0

Multi-Agent: Architecture Companion



basic



composite



analytical



AI architecture companion

Language, Technology & Architekturstil



📊 Language Distribution:

- **JavaScript:** 37.7% (frontend logic)
- **Java:** 29.6% (backend/server-side)
- **SCSS:** 13.5% (stylesheets)
- **CSS:** 9.3% (additional styling)
- **HTML:** 6.3% (web pages)
- **Gherkin:** 3.3% (BDD testing)
- **Shell:** 0.3% (build scripts)
- **SQL:** 0.1% (database queries)

🔧 Build Tools:

- **Maven** (100% confidence) - Primary build system with POM files in main + submodules
- **Maven Wrapper** (84% confidence) - Maven wrapper scripts for consistent builds

🍷 UI Framework Stack:

- **Bootstrap** (100% confidence) - Responsive CSS framework
- **jQuery** (100% confidence) - JavaScript library with DataTables & Peity plugins
- **Font Awesome** (70% confidence) - Icon library
- **Animate.css** (70% confidence) - CSS animation library
- **Normalize.css** (70% confidence) - CSS reset/normalization

🌐 Web Framework:

- **Spring Boot** (80% confidence) - Java web framework detected through imports and annotations

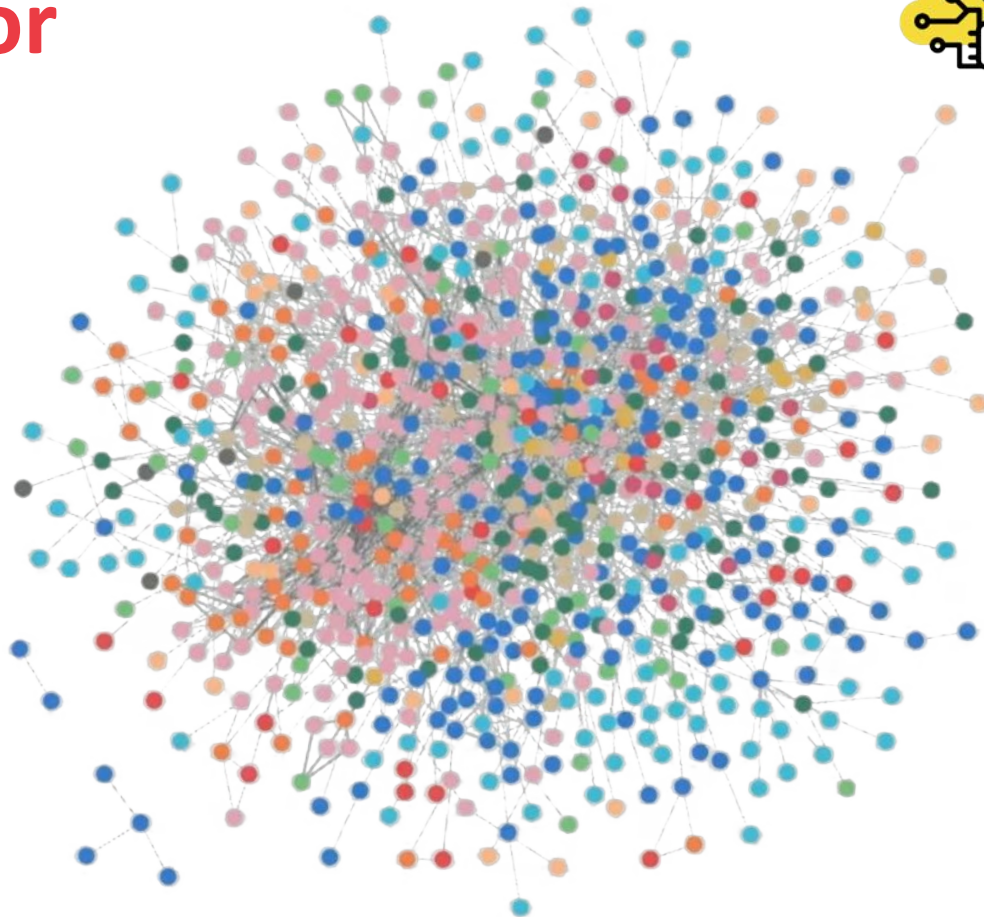
Key Insights:

1. **Multi-module Maven project** with separate **bank** and **credit** modules
2. **Spring Boot microservices architecture**
3. **Modern responsive web UI** using Bootstrap + jQuery
4. **Behavior-driven development** with Gherkin test scenarios
5. **Full-stack Java web application** with rich frontend

Graph Constructor



- Entitätsbeziehungen
- Methodenaufrufe
- Endpunkte
- Gemeinsame Änderung
- DB-Beziehungen
- Remote Methodenaufrufe
- Config-Dependencies
- Semantische Nähe
- ...



Gliederung und Datenfluss



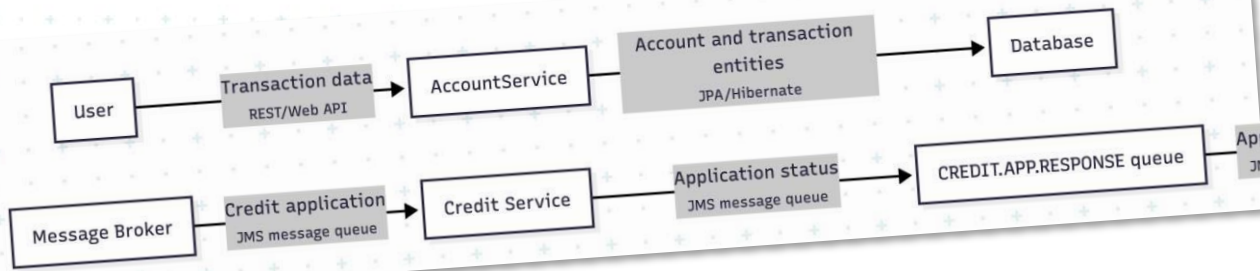
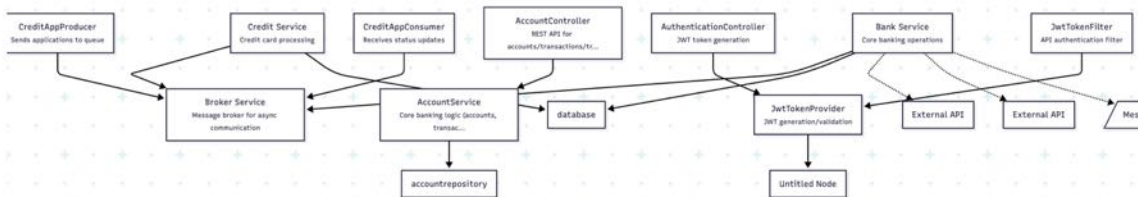
Projects / Personal / Untitled diagram

Code Auto-Update Docs X

```

1 ---
2 config:
3   layout: dagre
4 ---
5 flowchart TB
6   bank_service["Bank Service<br><small>Core ban
7   credit_service["Credit Service<br><small>Cred
8   accountcontroller["AccountController<br><small>
9   authenticationcontroller["AuthenticationContr
10  usercontroller["UserController<br><small>User
11  accountservice --> accountrepository["account
12  userservice --> userrepository["userrepositor
13  jwttokenfilter["JwtTokenFilter<br><small>API
14  creditappproducer["CreditAppProducer<br><small>
15  creditappconsumer["CreditAppConsumer<br><small>
16  bank_service --> open_banking_platform_obp_
17  jwttokenprovider --> n1["Untitled Node"]
18  obpservice["ObpService<br><small>Open Banking
19  visaservice["VisaService<br><small>VISA payme
20  searchservice["SearchService<br><small>ATM lo
21  multihttpsecurityconfig["MultiHttpSecurityCon
22  h2_database["Database"]]
23  mysql["Database"]]
24  postgresql["Database"]]
25  ms_sql_server["Database"]]
26  atm_location_service["External API"]
27

```

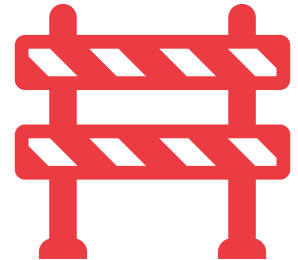


Guardrails

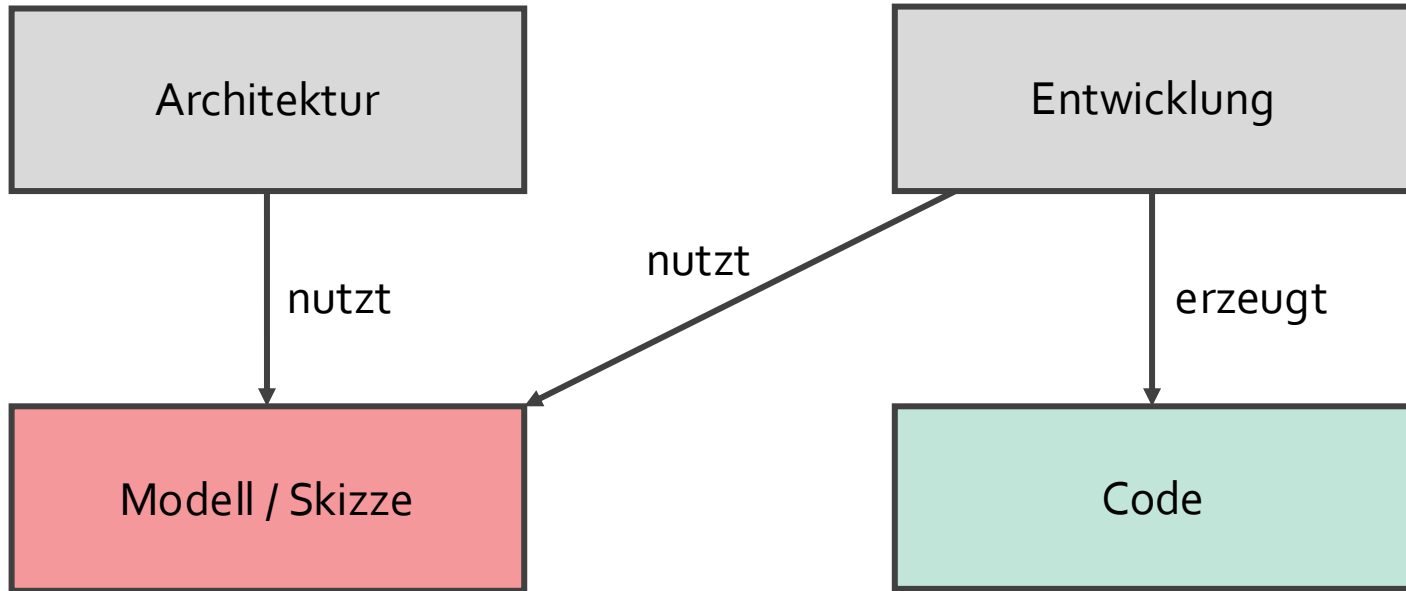
- **Preventative guardrails** are proactive guardrails that specify the outer bounds of what developers or agents can do.

- **Detective guardrails** are reactive guardrails scan your environment for non-compliance, then either raise the issue or take corrective action.

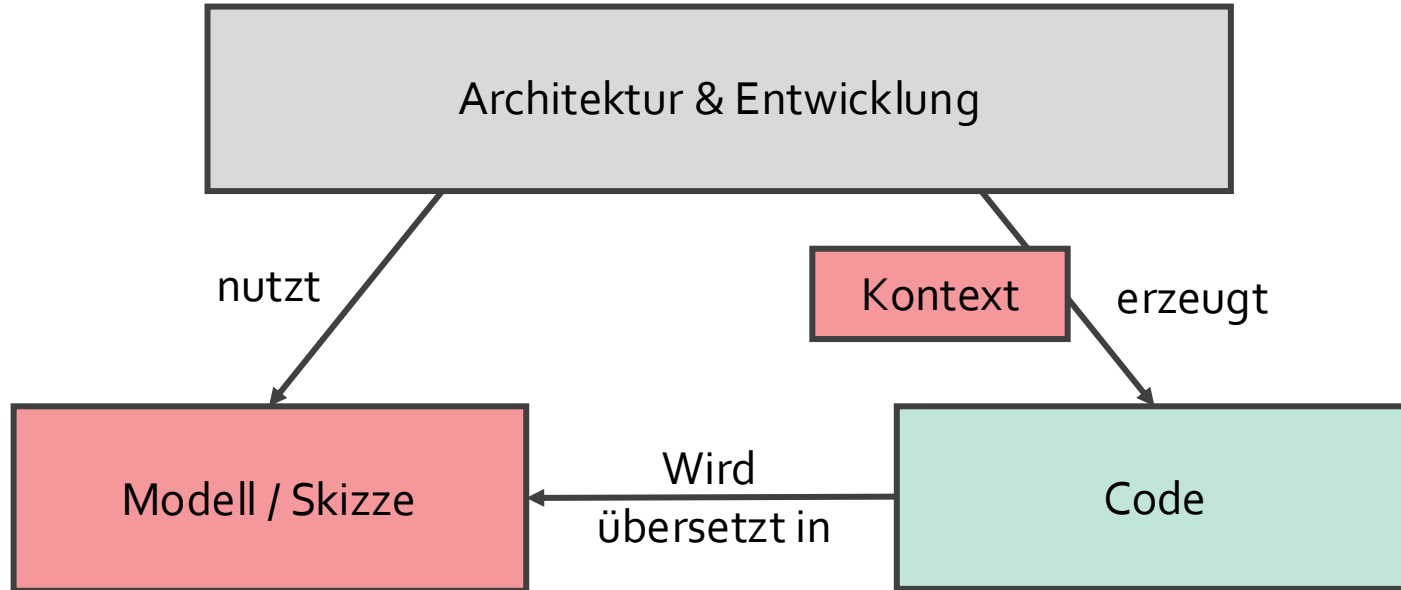
Klassisch, oder feingranularer mit „Hooks“ (Claude Code Hooks, Cursor Hooks, Azure SRE Agent, OpenAI Agent SDK, ...)



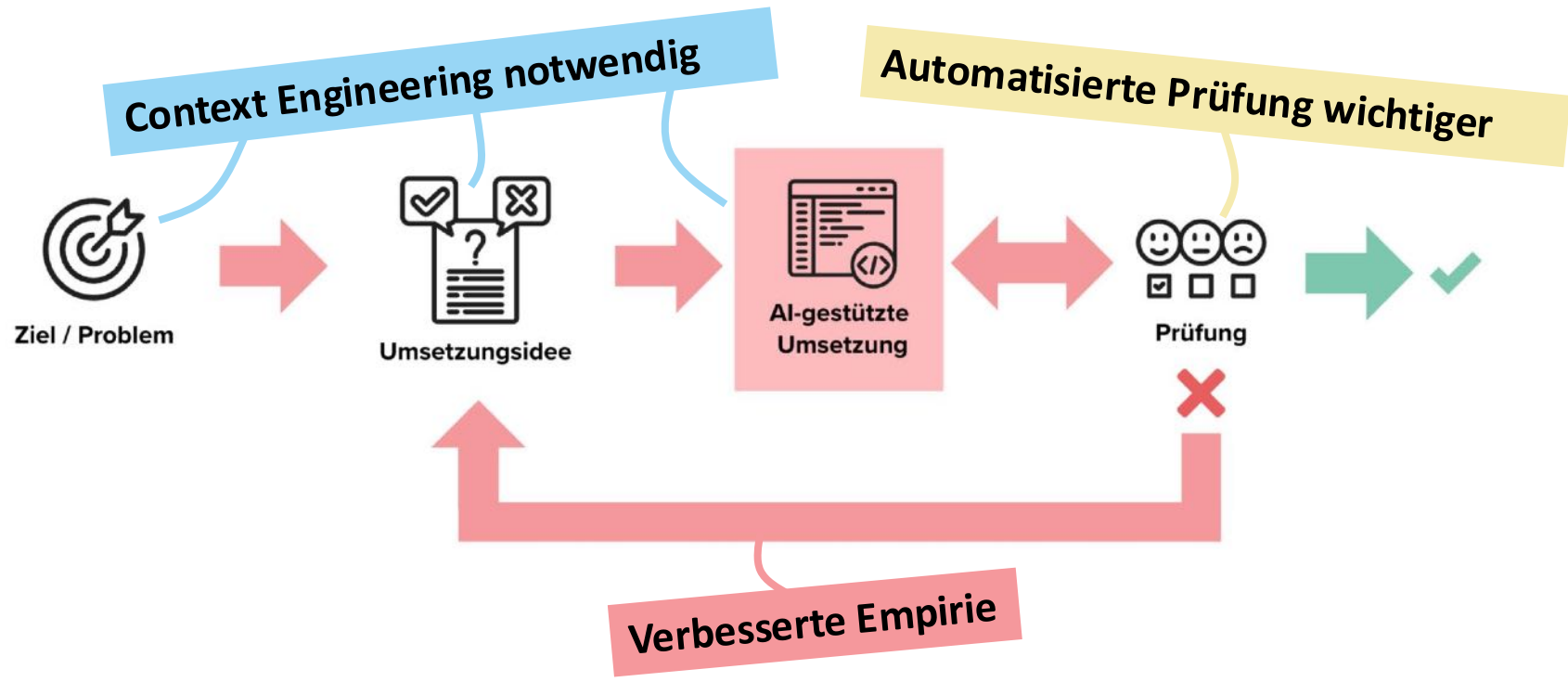
Klassisch: Architektur und Entwicklung



Agentic: Architektur und Entwicklung



Generative KI in der Entwicklung

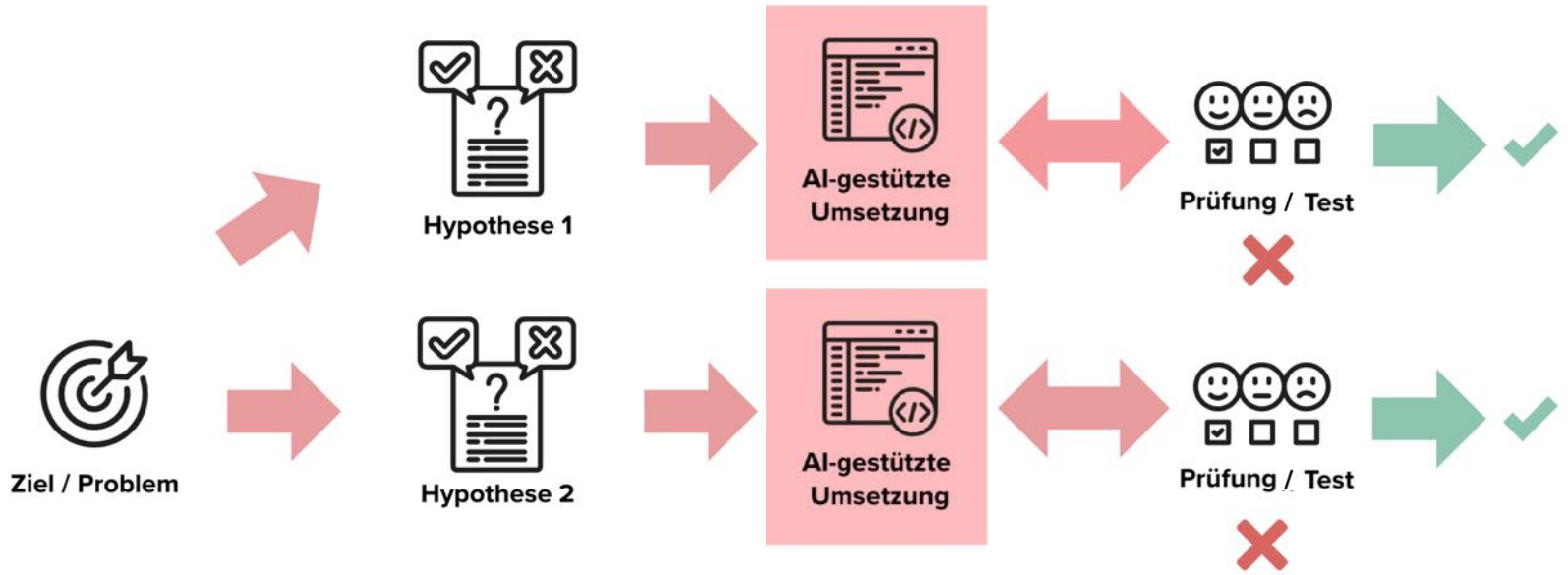




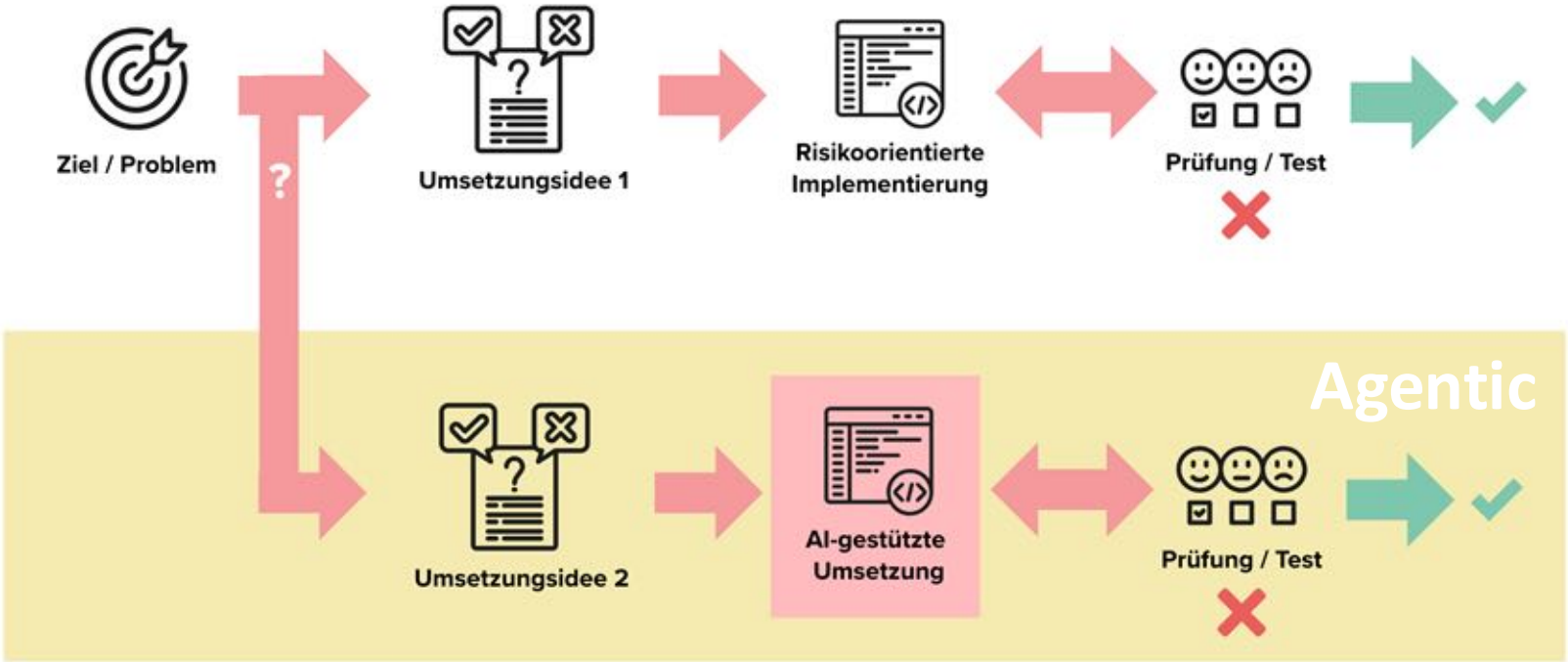
Verbesserte Empirie

Kontinuierlicher Lernprozess – individuell und in der Organisation

Hypothesen-orientierte Architektur



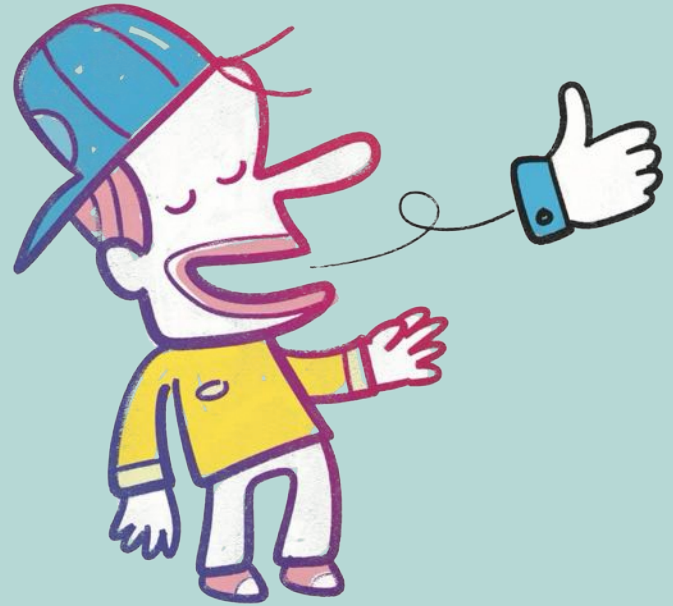
Agentic Coding professionell einführen



Agentic Architecture ist nicht gratis

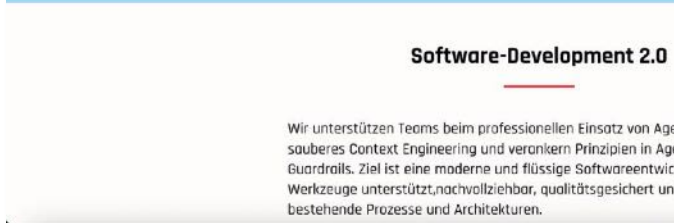
- Qualitätsziele kennen
- Rahmenbedingungen explizit machen
- Domäne und Strukturierung klar herausarbeiten
- Klare Schnittstellen schaffen
- Sprach-Eigenschaften und Best-Practices definieren
- Passende Testansätze schaffen
- Integration, Deployment und Monitoring professionalisieren
- Klare KI-Verwendung etablieren (inkl. Prozesse, Kontext etc.)
- ...

Feedback & Questions?





Folien & Infos auf embarc.de



AI-gestützte Architekturarbeit



Findet mich auf LinkedIn...

