kwcon 2026

AI and CCM – Practical use cases



Die Techniker

die Bayerische

die Mobiliar

kwsoft

AI and CCM – Real-World Scenarios

13,000,000,000



AI and CCM – Practical use cases



Andre Lohaus / Nico Lacroix / Tim Koster / Frank Schrader

Die Techniker
die Bayerische
die Mobiliar
kwsoft

From requirements to effective communication



Requirements

Prioritize needs

Development

Implement for reuse

Test

Ensure quality

Business Unit Benefit

More time for core tasks

Customer Benefit

Trust through clarity

Improve continuously

Evaluate real-world feedback and results. Evolve templates in a targeted way.

Die Techniker

die Bayerische

die Mobiliar

kwsoft

PHASE 1: DEVELOP TEMPLATE

1 **Requirements** — 2 Development — 3 Test

PHASE 2: USE TEMPLATE

4 Business Unit Benefit — 5 Customer Benefit

Understand needs, define goals, specify content

die Bayerische

Creating functional specifications from legacy code

Completeness check of requirements / identification of missing components in Jira/Confluence

Die Techniker

Quality check in the editorial process



PHASE 1: DEVELOP TEMPLATE

1 **Requirements** — 2 Development — 3 Test

PHASE 2: USE TEMPLATE

4 Business Unit Benefit — 5 Customer Benefit

Understand needs, define goals, specify content

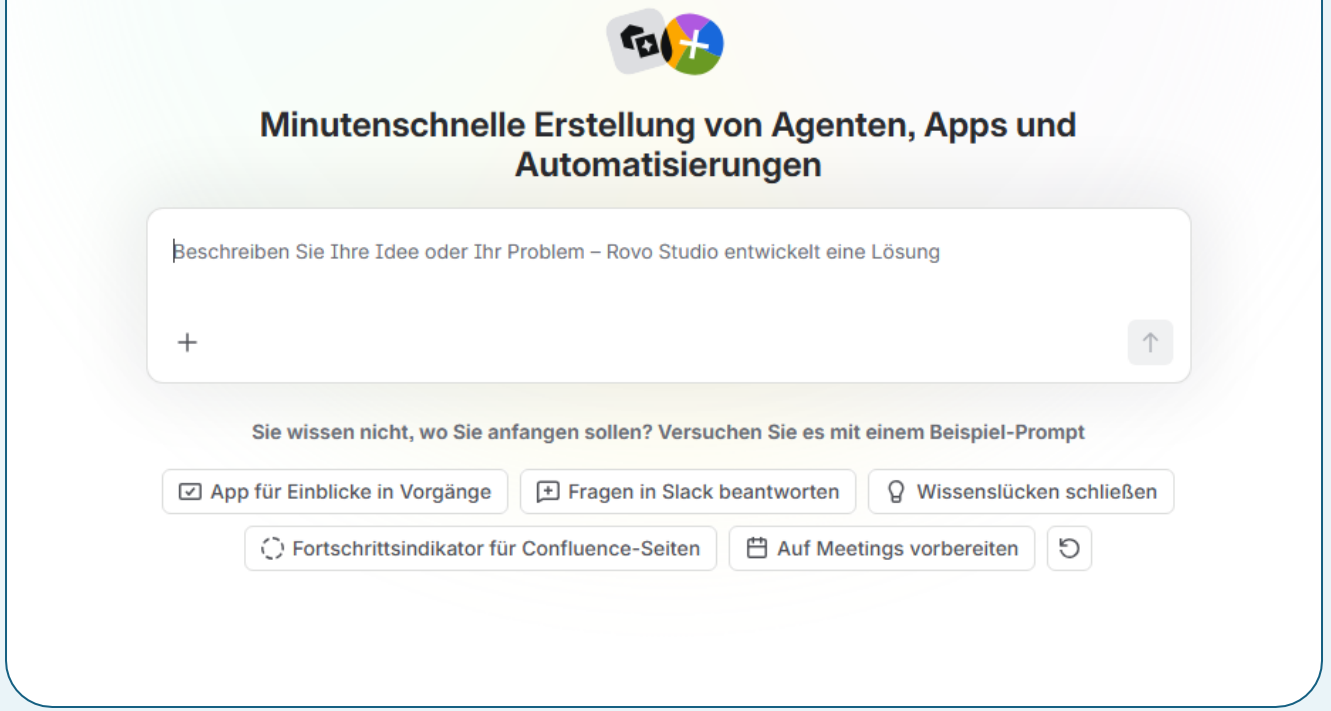
die Bayerische

Creating functional specifications from legacy code

Completeness check of requirements / identification of missing components in Jira/Confluence

Die Techniker

Quality check in the editorial process



PHASE 1: DEVELOP TEMPLATE

1 **Requirements** — 2 Development — 3 Test

PHASE 2: USE TEMPLATE

4 Business Unit Benefit — 5 Customer Benefit

Understand needs, define goals, specify content

die Bayerische

Creating functional specifications from legacy code

Completeness check of requirements / identification of missing components in Jira/Confluence

Die Techniker

Quality check in the editorial process



PHASE 1: DEVELOP TEMPLATE

1 Requirements — 2 **Development** — 3 Test

PHASE 2: USE TEMPLATE

4 Business Unit Benefit — 5 Customer Benefit

Design content and layout, integrate data and rules

die Bayerische

M/TEXT Tonic code generation

die Mobiliar

Template migration using Agentic Engineering

Text models with GitHub Copilot

Claude Code × M/TEXT TONIC

KI-Unterstützung in der Vorlagenentwicklung



PHASE 1: DEVELOP TEMPLATE

1 Requirements — 2 **Development** — 3 Test

PHASE 2: USE TEMPLATE

4 Business Unit Benefit — 5 Customer Benefit

Design content and layout, integrate data and rules

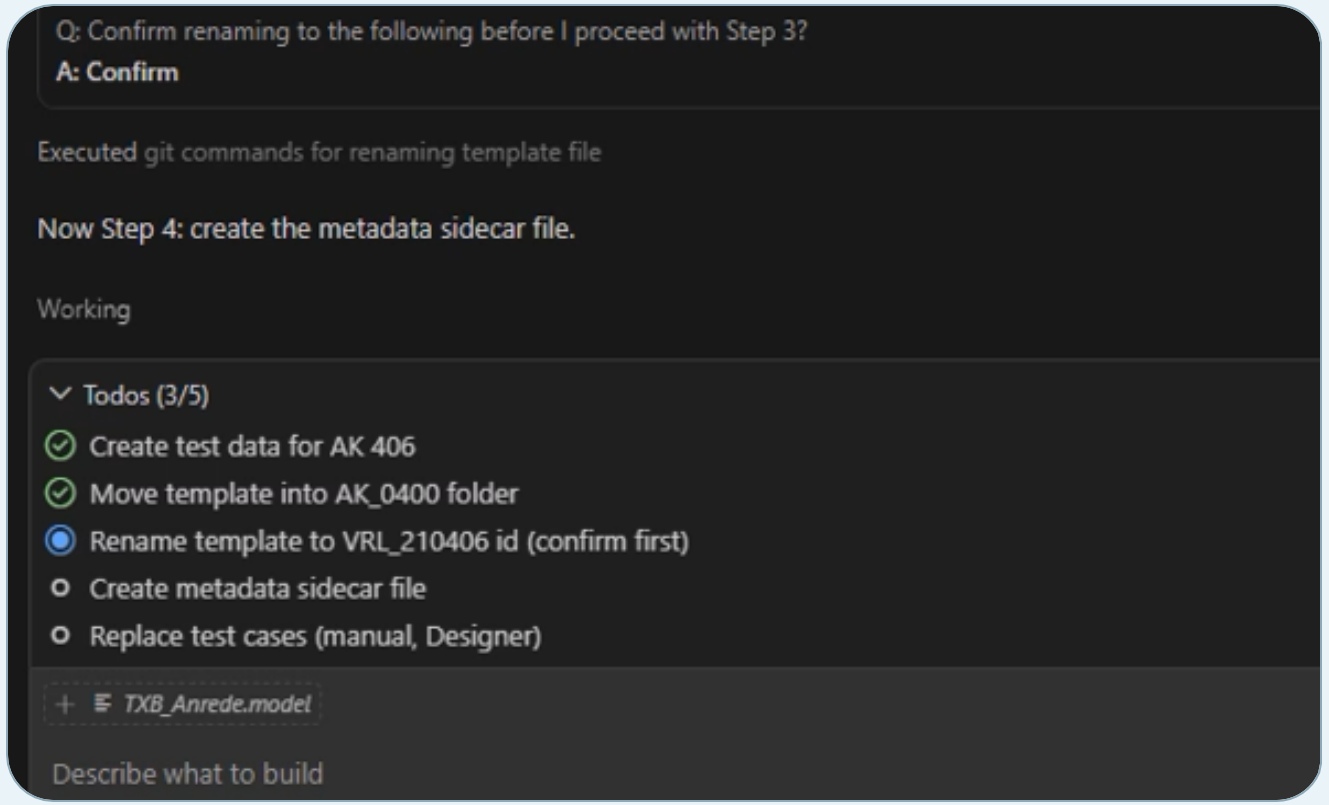
die Bayerische

M/TEXT Tonic code generation

die Mobiliar

Template migration using Agentic Engineering

Text models with GitHub Copilot



PHASE 1: DEVELOP TEMPLATE

1 Requirements — 2 **Development** — 3 Test

PHASE 2: USE TEMPLATE

4 Business Unit Benefit — 5 Customer Benefit

Design content and layout, integrate data and rules

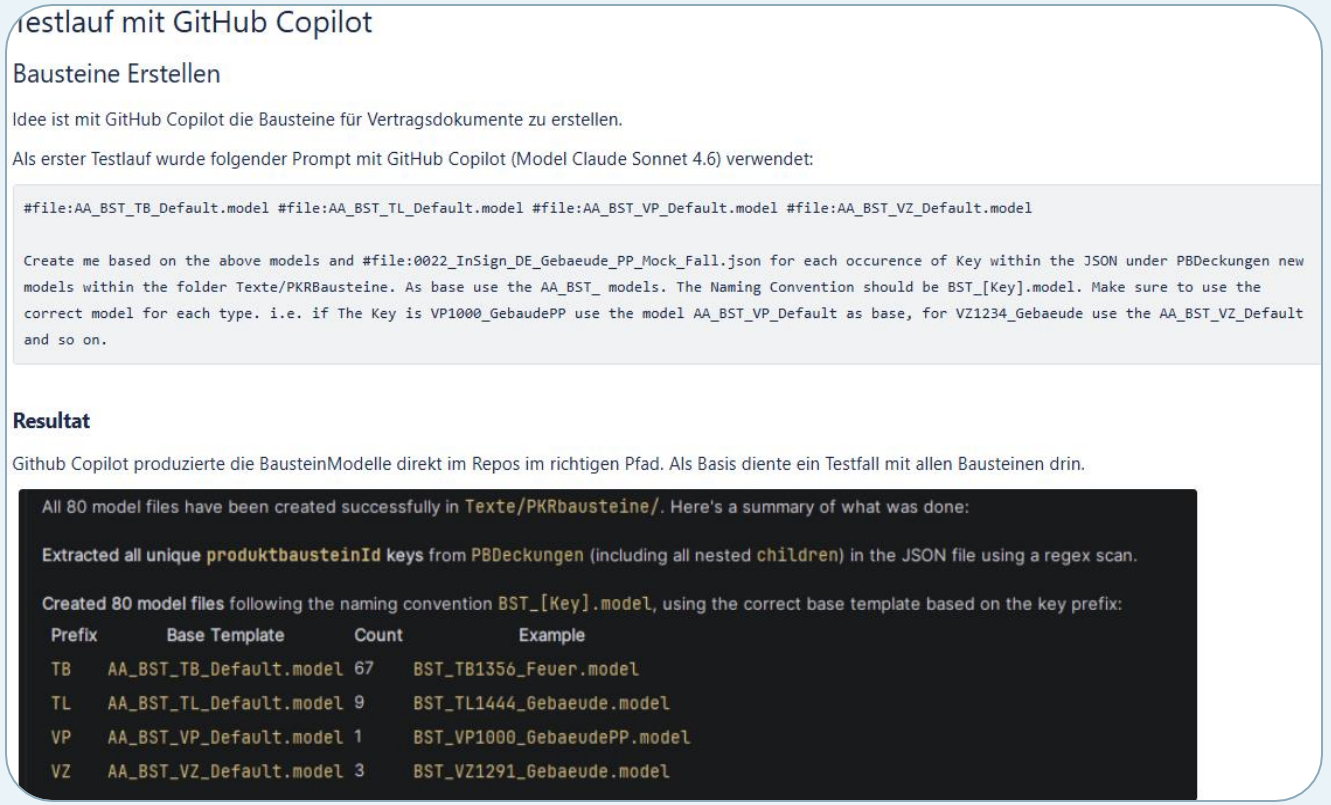
die Bayerische

M/TEXT Tonic code generation

die Mobiliar

Template migration using Agentic Engineering

Text models with GitHub Copilot



Testlauf mit GitHub Copilot

Bausteine Erstellen

Idee ist mit GitHub Copilot die Bausteine für Vertragsdokumente zu erstellen.

Als erster Testlauf wurde folgender Prompt mit GitHub Copilot (Model Claude Sonnet 4.6) verwendet:

```
#file:AA_BST_TB_Default.model #file:AA_BST_TL_Default.model #file:AA_BST_VP_Default.model #file:AA_BST_VZ_Default.model

Create me based on the above models and #file:0022_InSign_DE_Gebaeude_PP_Mock_Fall.json for each occurence of Key within the JSON under PBDeckungen new models within the folder Texte/PKRBausteine. As base use the AA_BST_ models. The Naming Convention should be BST_[Key].model. Make sure to use the correct model for each type. i.e. if The Key is VP1000_GebaudePP use the model AA_BST_VP_Default as base, for VZ1234_Gebaeude use the AA_BST_VZ_Default and so on.
```

Resultat

Github Copilot produzierte die BausteinModelle direkt im Repos im richtigen Pfad. Als Basis diente ein Testfall mit allen Bausteinen drin.

All 80 model files have been created successfully in `Texte/PKRbausteine/`. Here's a summary of what was done:

Extracted all unique `produktbausteinId` keys from `PBDeckungen` (including all nested `children`) in the JSON file using a regex scan.

Created 80 model files following the naming convention `BST_[Key].model`, using the correct base template based on the key prefix:

Prefix	Base Template	Count	Example
TB	AA_BST_TB_Default.model	67	BST_TB1356_Feuer.model
TL	AA_BST_TL_Default.model	9	BST_TL1444_Gebaeude.model
VP	AA_BST_VP_Default.model	1	BST_VP1000_GebaeudePP.model
VZ	AA_BST_VZ_Default.model	3	BST_VZ1291_Gebaeude.model

PHASE 1: DEVELOP TEMPLATE

1 Requirements — 2 Development — 3 **Test**

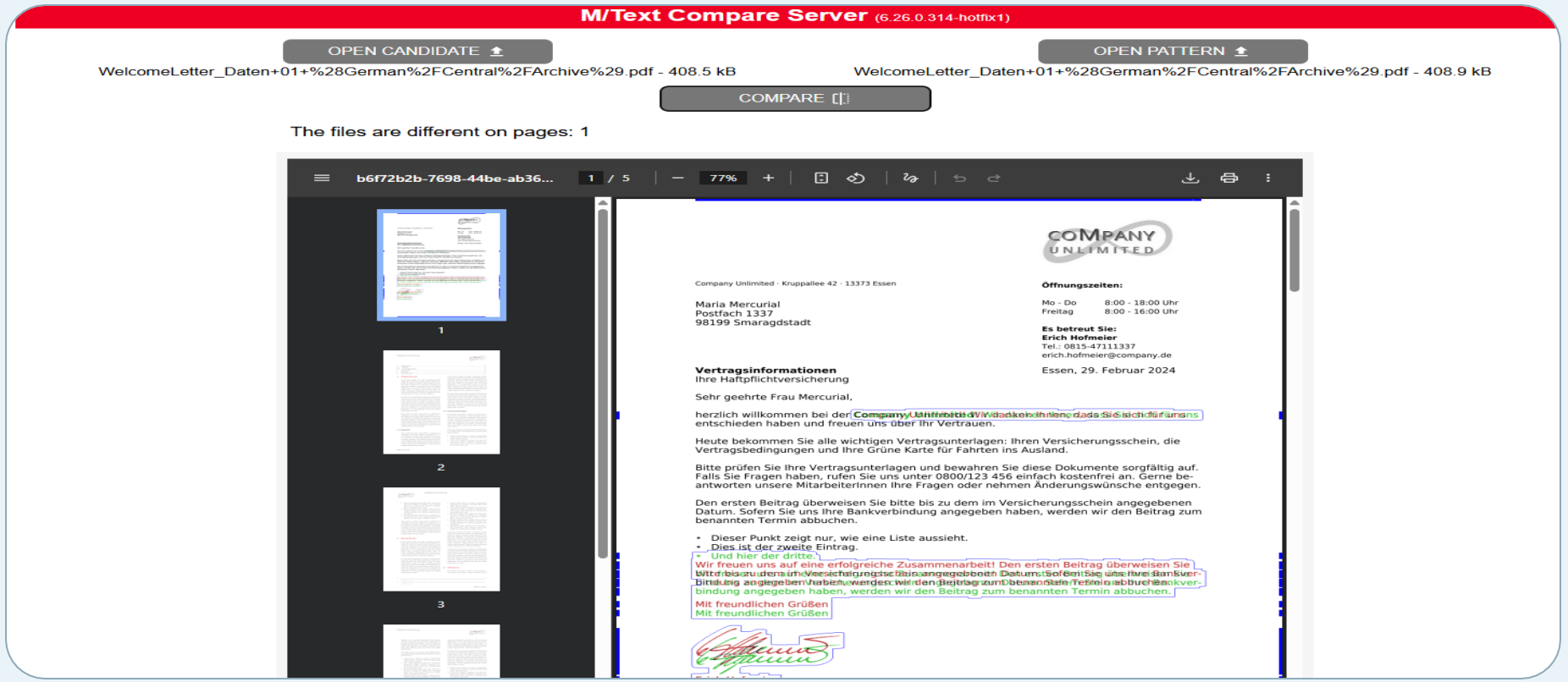
PHASE 2: USE TEMPLATE

4 Business Unit Benefit — 5 Customer Benefit

Ensure quality

kwsoft

Test case generation with AI tools, functional QA and regression tests



Die Techniker — die Bayerische — die Mobiliar — kwsoft

PHASE 1: DEVELOP TEMPLATE

1 Requirements — 2 Development — 3 Test

PHASE 2: USE TEMPLATE

4 **Business Unit Benefit** — 5 Customer Benefit

More time for core tasks

Automated text creation

Quality check during letter creation

Internal summary of the sent letter



PHASE 1: DEVELOP TEMPLATE

1 Requirements — 2 Development — 3 Test

PHASE 2: USE TEMPLATE

4 **Business Unit Benefit** — 5 Customer Benefit

More time for core tasks

Automated text creation

Quality check during letter creation

Internal summary of the sent letter



PHASE 1: DEVELOP TEMPLATE

1 Requirements — 2 Development — 3 Test

PHASE 2: USE TEMPLATE

4 **Business Unit Benefit** — 5 Customer Benefit

More time for core tasks

Automated text creation.

Quality check during letter creation

Internal summary of the sent letter



Trust through clarity

Letter summary for customers within the letter



AI and CCM – Practical use cases



Andre Lohaus / Nico Lacroix / Tim Koster / Frank Schrader

Die Techniker
die Bayerische
die Mobiliar
kwsoft

kwcon 2026

AI and CCM – Practical use cases



Thank you!

Die Techniker

die Bayerische

die Mobiliar

kwsoft