

"Handmade" Slides

Wie moderne KI die klassische Softwareentwicklung transformiert



Mischa Hüschen
*Managing Director
prodot GmbH*



Christian Kratky
*Principal Software Engineer
prodot GmbH*



Stefan Wilkes
*Unit Lead / Senior Software Architect
prodot GmbH*

Agenda

Generative KI

... und ihre Varianten

“Mainstream” Software

Unternehmenssoftware

Demo Time



Wie moderne KI die klassische Softwareentwicklung transformiert

A large, abstract sculpture in a courtyard, overlaid with a red tint and the text 'Generative KI'. The sculpture is composed of several interconnected, rounded forms, possibly representing a stylized figure or a complex structure. The background shows a modern building with many windows, suggesting an urban or institutional setting. The overall image has a strong red color cast.

Generative KI

ChatGPT

Generative KI

Algorithmen, die aus Daten lernen und neue Daten generieren können, die den trainierten Daten ähneln

Generative KI vs Diskriminative KI

- *Generative Modelle lernen und generieren neue Daten*
- *Diskriminative Modelle klassifizieren und treffen Vorhersagen*

z.B. Large Language Models (LLMs)

- *KI-Modelle, die natürliche Texte aus großen Datenmengen generieren*
- *Verwendung von tiefen neuronalen Netzen, z.B. mit Transformers*
- *Fähigkeiten: Klassifizierung, Zusammenfassung, Übersetzung, Generierung und Dialog*
- *Beispiele für LLMs: OpenAI (Microsoft) GPT, Google Bard, Meta Llama*

ChatGPT everywhere

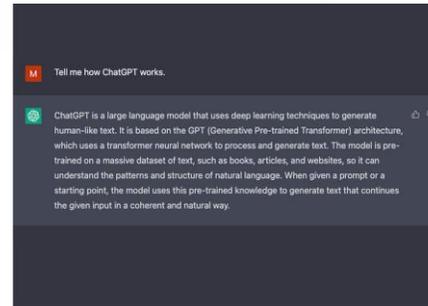


Wozu generative KI?

Generative KI spielt eine Schlüsselrolle bei der Schaffung neuer Inhalte und der Automatisierung von Aufgaben

Beispiele für generative KI

- Textgenerierung
ChatGPT
- Musikgenerierung
Google MusicLM
- Bildgenerierung
Midjourney
- 3D-Generierung
Luma AI
- Games: NPCs
**Generative Agents:
Interactive Simulacra of
Human Behavior**
- und mehr ...



Google
MusicLM



MIDJOURNEY

"marble statue of a greek philosopher"

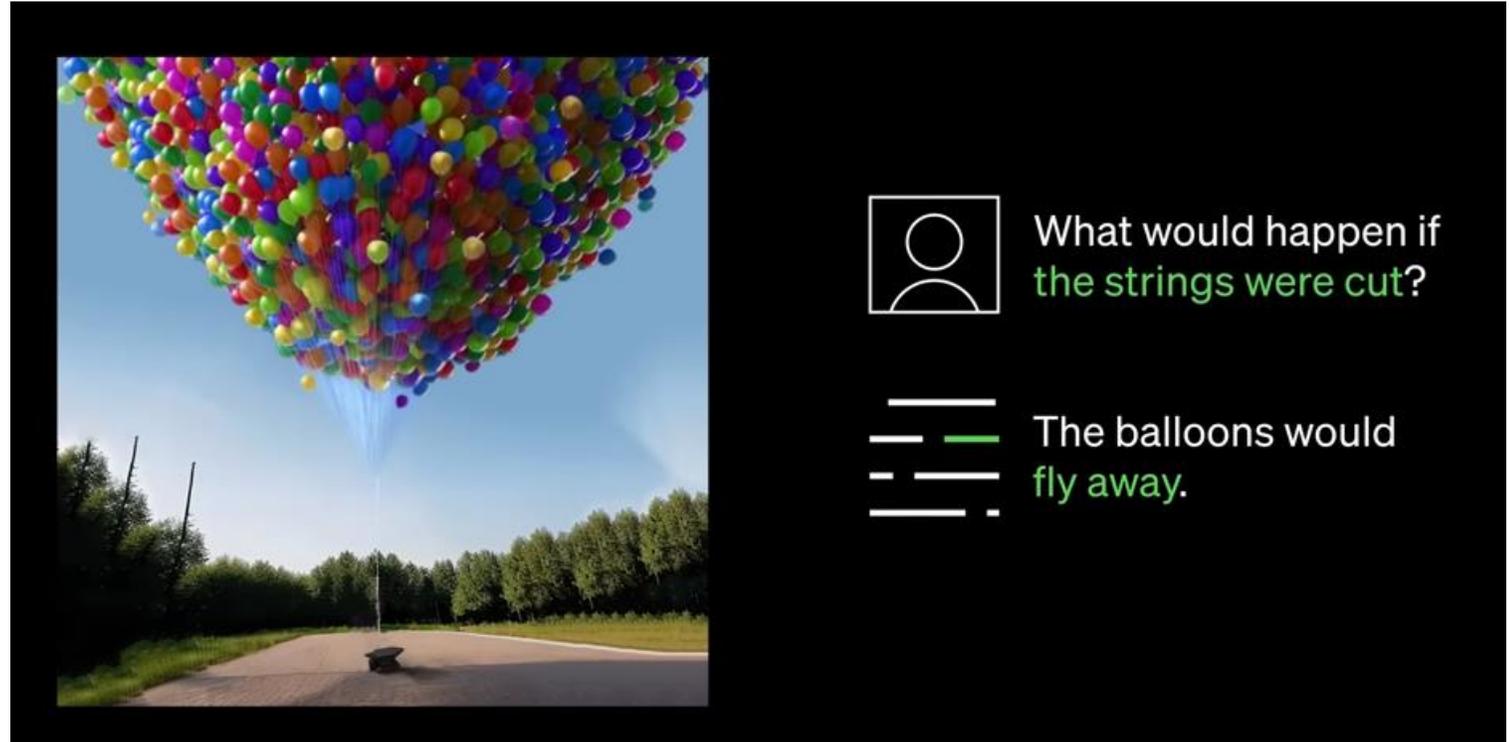


Multimodale KI-Modelle



*... bezieht sich auf die Integration mehrerer Modalitäten wie **Text, Bild, Sprache, Video etc.** in ein einziges KI-Modell.*

- *unterschiedliche Datenformate*
- *Bildbeschreibung / -analyse*
- *Potenzial für verbesserte User-Interaktion*



Verständnis von Kontext!

Hollywood Schauspieler und Autoren streiken

- KI schreibt Drehbücher
- KI erzeugt fertige Filmszenen auf Basis von digitalisierten Schauspielern
- Sie wollen nicht “überflüssig” werden!

Was hat das mit Software zu tun?

HOLLYWOOD SCREENWRITERS V



KI bei der Softwareerstellung

Generative KI kann auch Source Code

- ... erzeugen
- ... verstehen und erklären
- ... verbessern
- ... beim Debuggen unterstützen
- ... auf Sicherheit überprüfen
- ... auf Rechte prüfen



Was bedeutet das für die Softwareentwicklung?

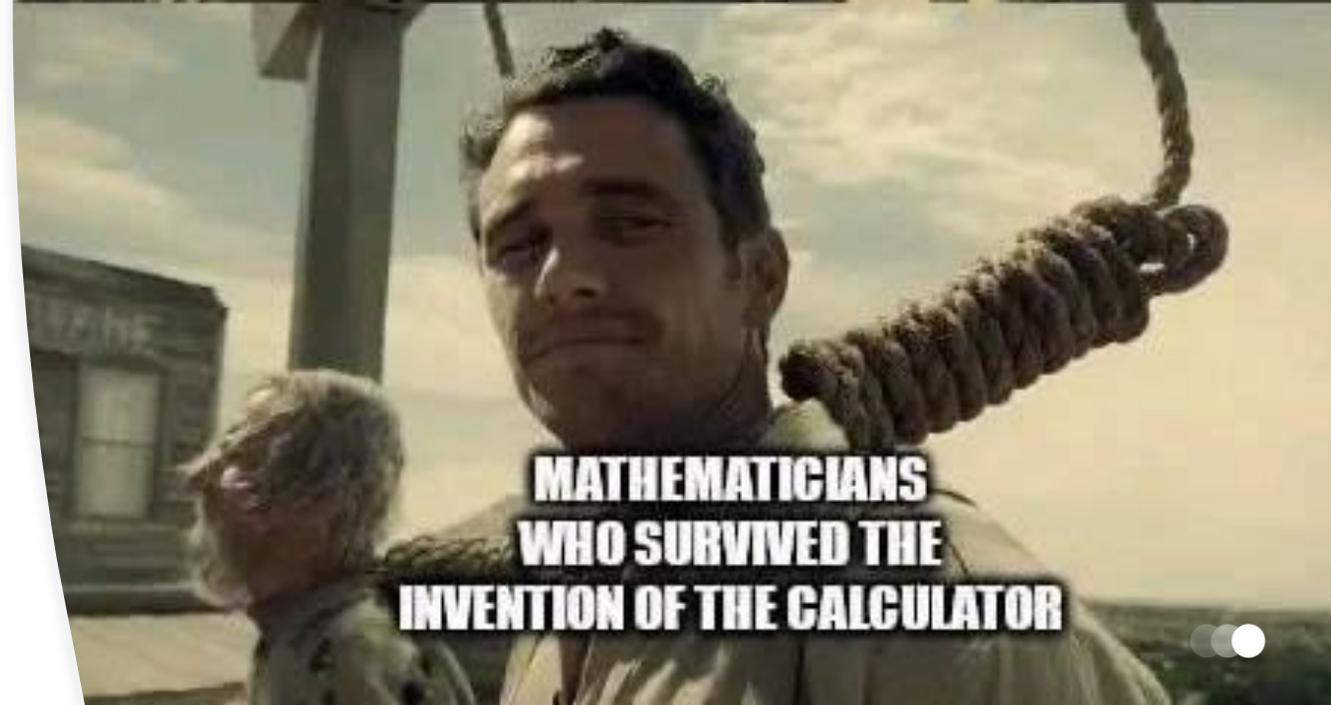
Production-Software ist komplex und kombiniert Dinge wie

- Architektur, Funktionalität, Design
- Implementierung, Testing
- Maintenance, DevOps
- ... und mehr

Softwareentwickler designen und erstellen Lösungen, die

- aus Bedürfnissen von Kunden und Benutzern
- und den spezifischen Rahmenbedingungen der gewünschten Lösung
- viele weitere Faktoren in Einklang bringen
- und betreiben, warten und supporten diese.

Aber die Art und Weise, wie wir das tun, ändert sich!



Versprechen von generativer KI bei der Softwareerstellung

- **Effizienzsteigerung**
KI-gestützte Tools reduzieren repetitives Codieren und erhöhen die Produktivität.
- **Qualitätsverbesserung**
Bessere Codevorschläge und Fehlererkennung erhöhen die Softwarequalität.
- **Lernunterstützung**
Unterstützt das Erlernen neuer Sprachen oder Frameworks.
- **Individualisierte Vorschläge**
KI lernt aus dem individuellen Code-Stil und passt Vorschläge an.
- **Zeitersparnis**
Weniger Zeit mit der Suche nach Lösungen im Web oder in der Dokumentation.
- **Rechteevaluierung**
Rechte und Lizenzen von KI-Code und Code-Fragmenten können in Zukunft geprüft und bewusst eingesetzt werden.

Es ist noch lange nicht perfekt aber es entwickelt sich.

... und was macht generative KI mit der Software selbst?

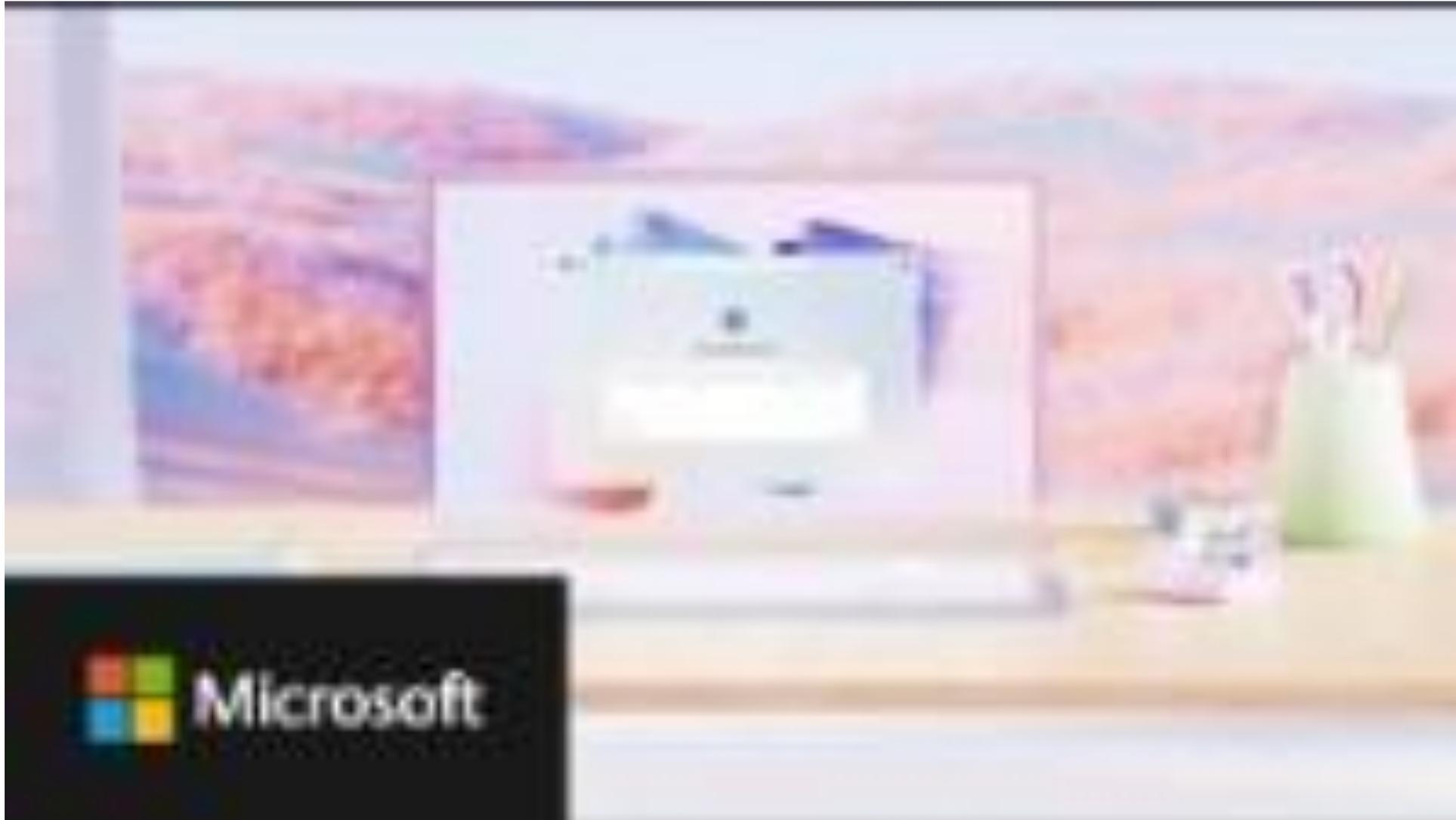


Wie moderne KI die klassische Softwareentwicklung transformiert



... in „Mainstream“ Software

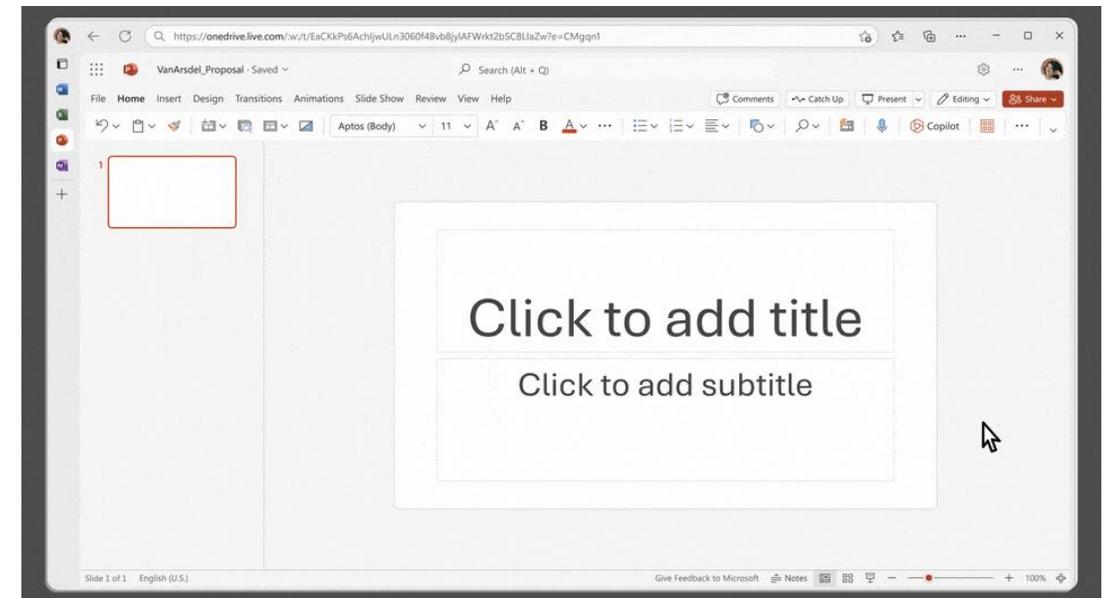
Introducing Microsoft 365 Copilot



Microsoft Copilot

... wird in die gesamte Microsoft Produktpalette integriert

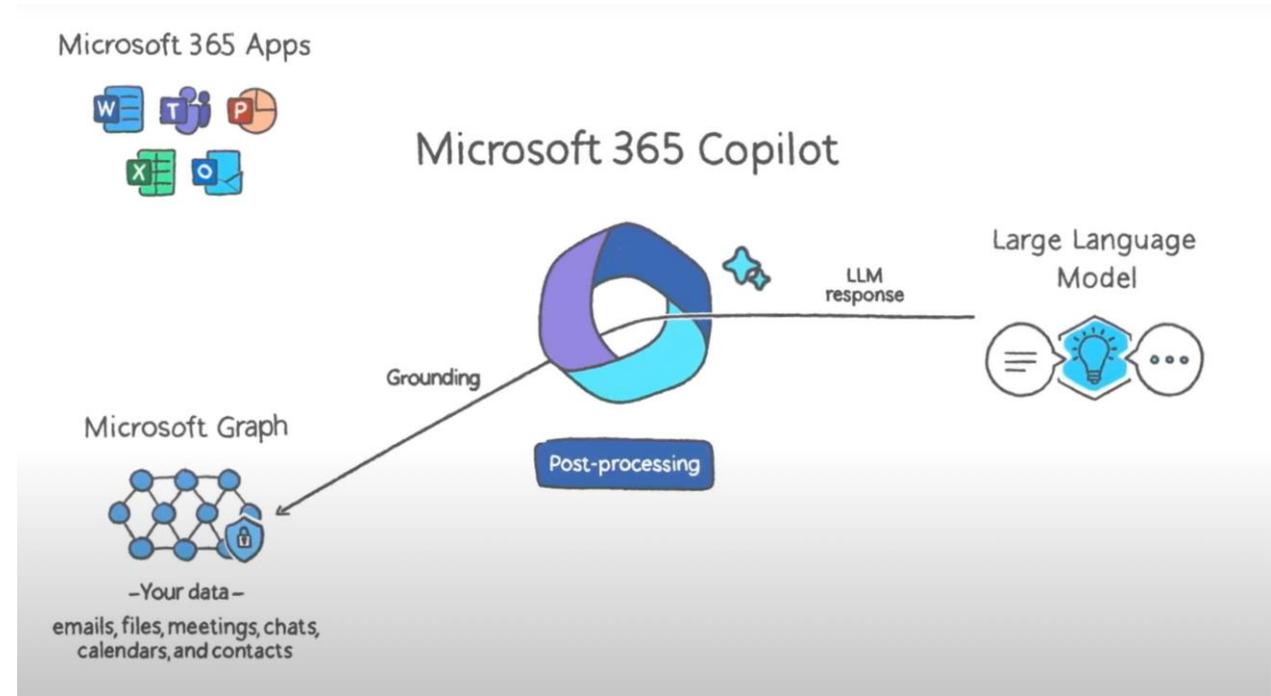
- *Windows*
- *Outlook*
- *Word*
- *PowerPoint*
- *Excel*
- *Viva Suite*
- *PowerApps und PowerApps Designer*
- *PowerBI*
- *Security Copilot*
- *uvm.*



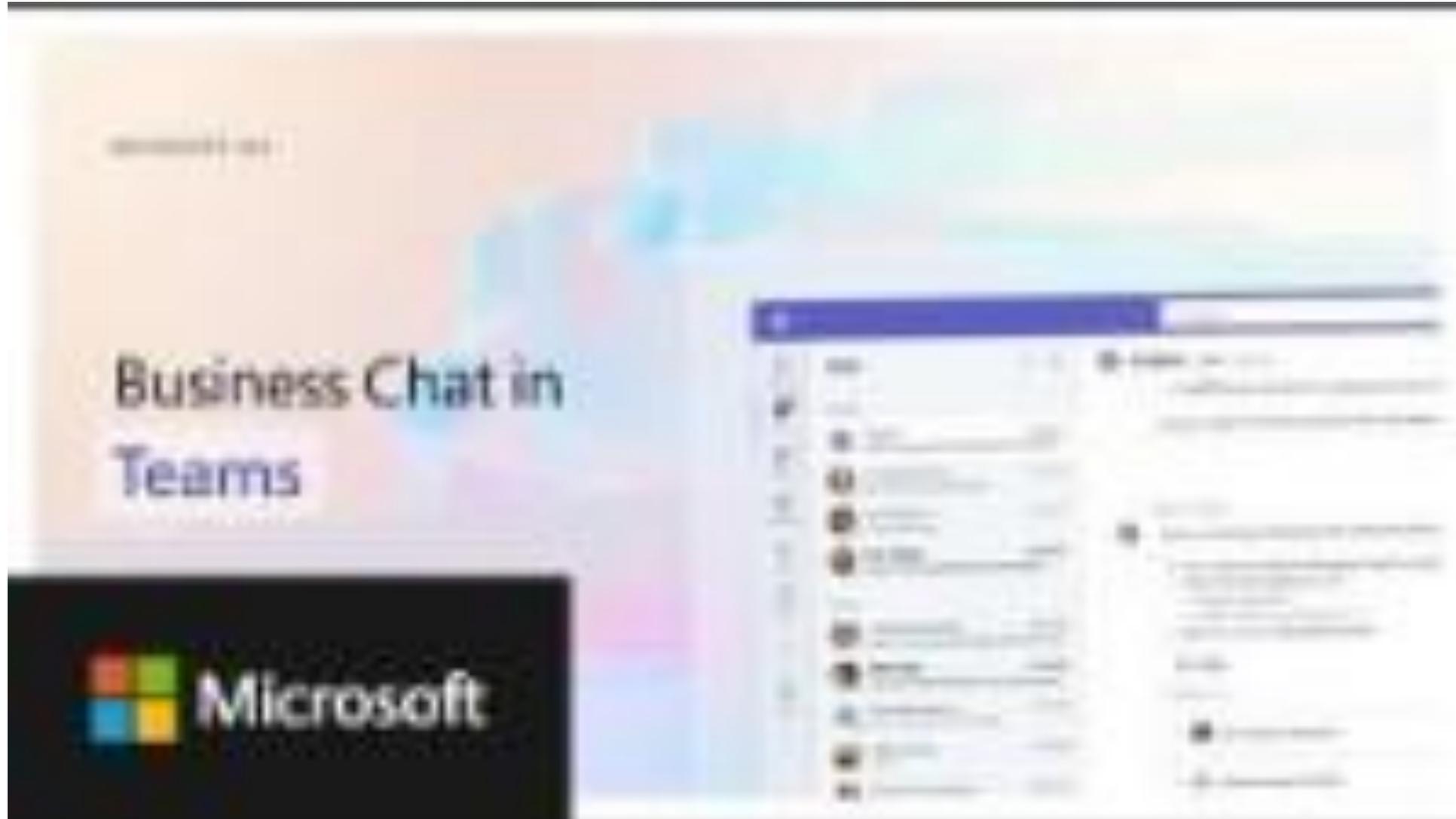
M365 Zugriff durch den Copilot

Zugriff auf die Daten des M365 Mandanten

- *Teams Chats und Meetings*
"Business Chat"
- *SharePoint*
- *Dokumente, Excel Sheets, PowerPoints etc.*
- *Loop*
- *E-Mails, Kalender, Kontakte*
- *Viva Engage*
- *Microsoft Dataverse*
- *uvm.*



Business Chat in Microsoft Teams



Software-Transformation | "Mainstream" Software

Country	Customer	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Discounts	Sales	COGS
France	VanArndel	Biodegradable Cleaning Products	Low	4345	\$10	\$352	\$1,529,640	\$15,294	\$1,514,146	\$
USA	VanArndel	Biodegradable Cleaning Products	Low	4245	\$10	\$352	\$1,494,240	\$14,942	\$1,479,298	\$
Australia	VanArndel	Biodegradable Cleaning Products	None	4124	\$10	\$352	\$1,451,648	\$0	\$1,451,648	\$
Mexico	VanArndel	Biodegradable Cleaning Products	Medium	3954	\$10	\$352	\$1,391,808	\$83,508	\$1,308,300	\$
France	Contoso Pharmaceuticals	Biodegradable Cleaning Products	Low	4902	\$10	\$127	\$628,904	\$6,289	\$622,615	\$
France	Contoso Pharmaceuticals	Recycled Can	Low	4903	\$5	\$127	\$622,681	\$6,227	\$616,454	\$
Canada	Proseware	Natural Personal Care Products	Low	1954	\$270	\$302	\$590,108	\$5,901	\$584,207	\$
Canada	Proseware	Biodegradable Cleaning Products	High	1990	\$220	\$302	\$600,980	\$84,137	\$516,843	\$
USA	VanArndel	Compostable Food Packaging	Low	1254	\$3	\$380	\$476,520	\$4,765	\$471,755	\$
USA	Proseware	Natural Personal Care Products	Medium	1465	\$224	\$302	\$442,430	\$26,546	\$415,884	\$
USA	Proseware	Sustainable Snack Packaging	Medium	1410	\$120	\$302	\$425,820	\$25,549	\$400,271	\$
Australia	Proseware	Natural Personal Care Products	Medium	1369	\$224	\$302	\$413,438	\$24,806	\$388,632	\$
USA	VanArndel	Compostable Food Packaging	Medium	1052	\$3	\$380	\$399,760	\$23,986	\$375,774	\$
Canada	VanArndel	Compostable Snack Packaging	Low	980	\$112	\$380	\$372,400	\$3,724	\$368,676	\$
Australia	Proseware	Sustainable Snack Packaging	Medium	1298	\$120	\$302	\$391,996	\$23,520	\$368,476	\$
Australia	VanArndel	Compostable Food Packaging	None	954	\$3	\$380	\$362,520	\$9	\$362,520	\$
Canada	Contoso Pharmaceuticals	Biodegradable Cleaning Products	Low	2785	\$110	\$127	\$353,695	\$3,537	\$350,158	\$

Chat

Copilot Chat About

Good morning Kati

Here are some ideas that might help you get started:

- Get a summary of everything you need to catch up on, including meetings
- Find and use info that's buried in documents or lost in conversations
- Create content with all of it—drawing from your files, conversations, events, topics, and more

As your Copilot, I'm here to assist you but I do make mistakes, so sources are provided for your review when possible.

Click to add title

Click to add subtitle

Create content with Copilot

A job description for... 10 ideas for... A press release for... A project proposal about...

Describe what you'd like to create, and Copilot will generate a draft

Aktuelles Preisschild: \$30 / User / Monat



Der “iPhone Moment”

... hatte 2007 einen tiefgreifenden Einfluss auf die Entwicklung von Software und Hardware.

- *Es wurden neue “UI/UX Standards” gesetzt*

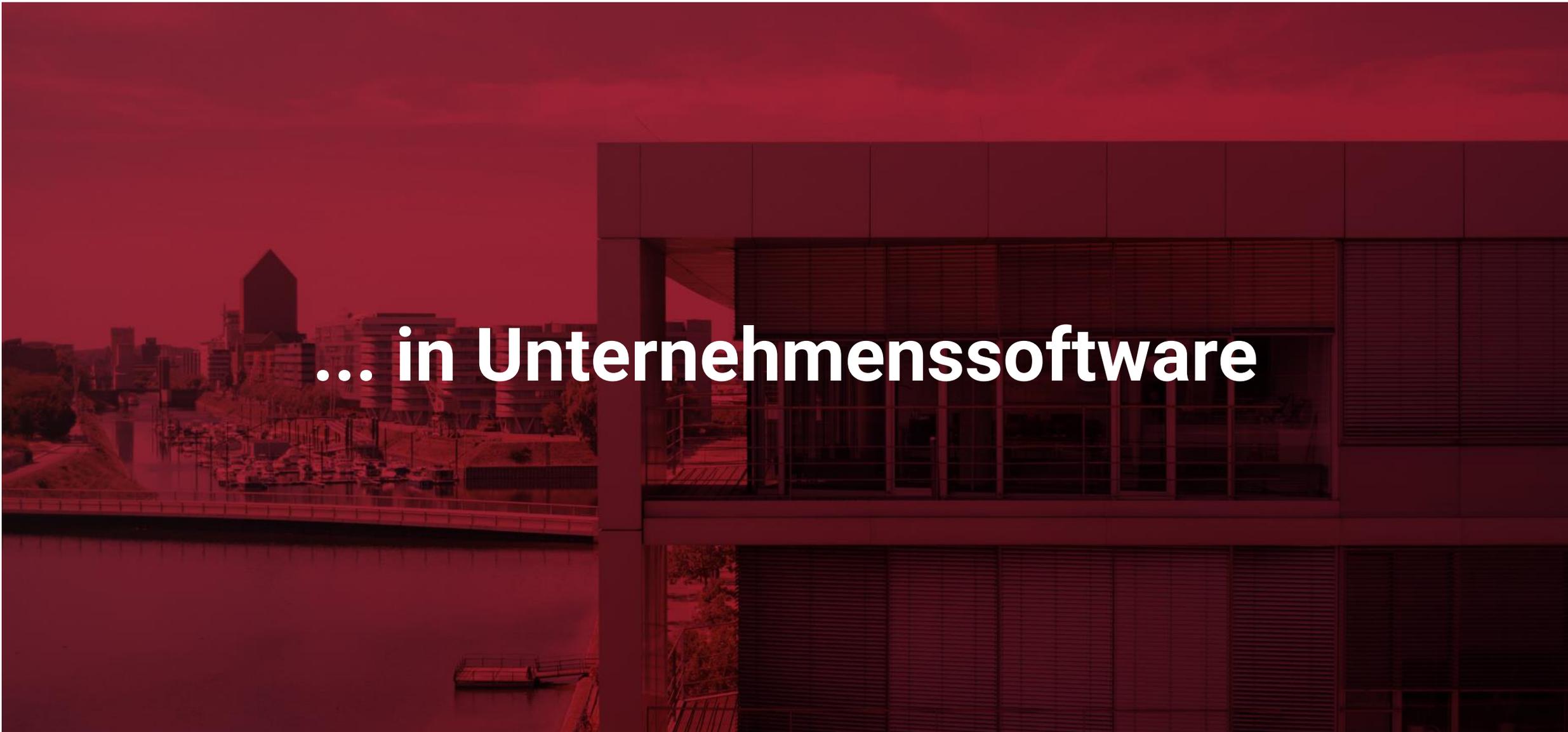
User werden KI-Unterstützung in Zukunft erwarten!

- ***Spoiler:***
Es dauert diesmal nicht so lange wie beim iPhone

“Was bedeutet das für Unternehmensoftware?”

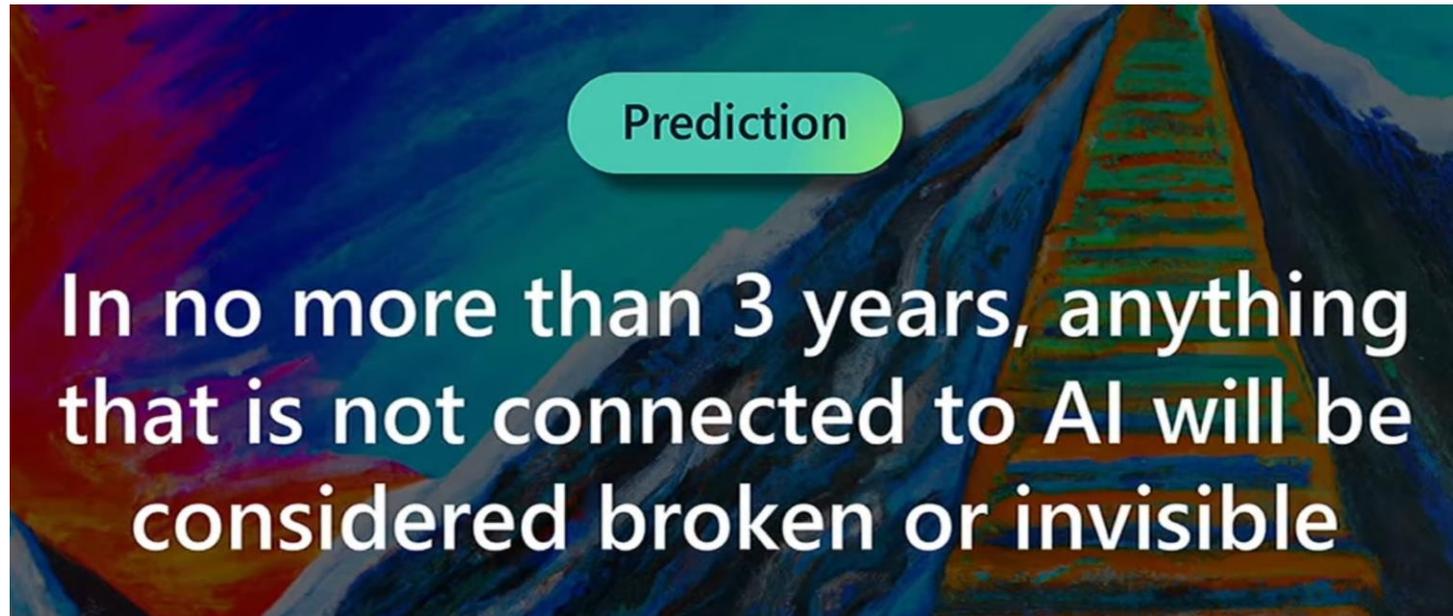


Wie moderne KI die klassische Softwareentwicklung transformiert



... in Unternehmenssoftware

Wo führt das hin?



Microsoft Build, Mai 2023



Sam Schillace, CVP, Deputy CTO, Microsoft

*Mitverantwortlich für die MS Copilot
Implementierung
ehemals Box, Google etc.*

“Ist das ein realistisches Szenario?”

Beispiele für KI-Unterstützung in Softwaresystemen

- *KI hat bisher hauptsächlich die Daten und Ergebnisse von Software*
 - *... beeinflusst*
 - *... verbessert*
 - *... möglich gemacht*
- **Datenanalysen / -vorhersagen**

“klassische” Verarbeitung von Bestandsdaten oder Vorhersagen auf deren Basis

z.B. Predictive Maintenance

Beispiele für KI-Unterstützung in Softwaresystemen

- *Jetzt treibt sie ...*
 - *neue Formen der Interaktion mit Menschen, Software und Daten*
 - *Verständnis von Inhalten und den Bezug untereinander*
- **“Reasoning”**
 - *„Mentaler“ Prozess, logische Schlüsse zu ziehen und Vorhersagen aus vorhandenem Wissen, Fakten und Überzeugungen zu machen.*
 - *„Reasoning“ ist eine Methode, um Fakten aus bestehenden Daten abzuleiten.*
 - ➔ *Assistenzsysteme wie die Microsoft Copilot-Systeme*
 - ➔ *Verständnis von Text (auch gesprochenem), Bild, Video, Audio ...*

Es ist wichtiger, worauf das Sprachmodell zugreifen kann, als was es bereits kennt!

“Reasoning” durch Semantische Programmierung

	Syntax	Semantik
Natürliche Sprache	<i>Regeln und Muster, wie Wörter und Sätze gebildet und strukturiert werden.</i>	<i>Bedeutung und Interpretation von Wörtern und Sätzen und wie sie sich auf die reale Welt beziehen.</i>
Programmiersprachen	<i>Regeln und Symbole, wie Code und Befehle geschrieben und strukturiert werden.</i>	<i>Bedeutung und Ausführung von Code und Befehlen und wie sie den Systemzustand beeinflussen.</i>
Neues Mindset	<i>... weg von präzise und zerbrechlich</i>	<i>... hin zu umfassend und flexibel</i>

Wie können wir den Mehrwert von Sprachmodellen bezogen auf diese Flexibilität mit unseren eigenen Daten und Ressourcen nutzbar machen?

Semantische Programmierung

- *Wie benutzt man ein Sprachmodell in eigener Software?*
- *Wer interpretiert den Wunsch des Users?*

- *Einen **“Copilot”** baut man nicht nur durch Aufrufen der API eines Sprachmodells*
- *Die Software orchestriert den **“Denkprozess”** und stellt Zugriff auf Daten und Aktionen zur Verfügung*
- **Der User gibt ein Ziel vor, die KI ermittelt eine Strategie zur Erreichung!**

Die Software “plaudert” im Hintergrund mit dem Large Language Model über das Problem des Users!

„Tree of Thought“

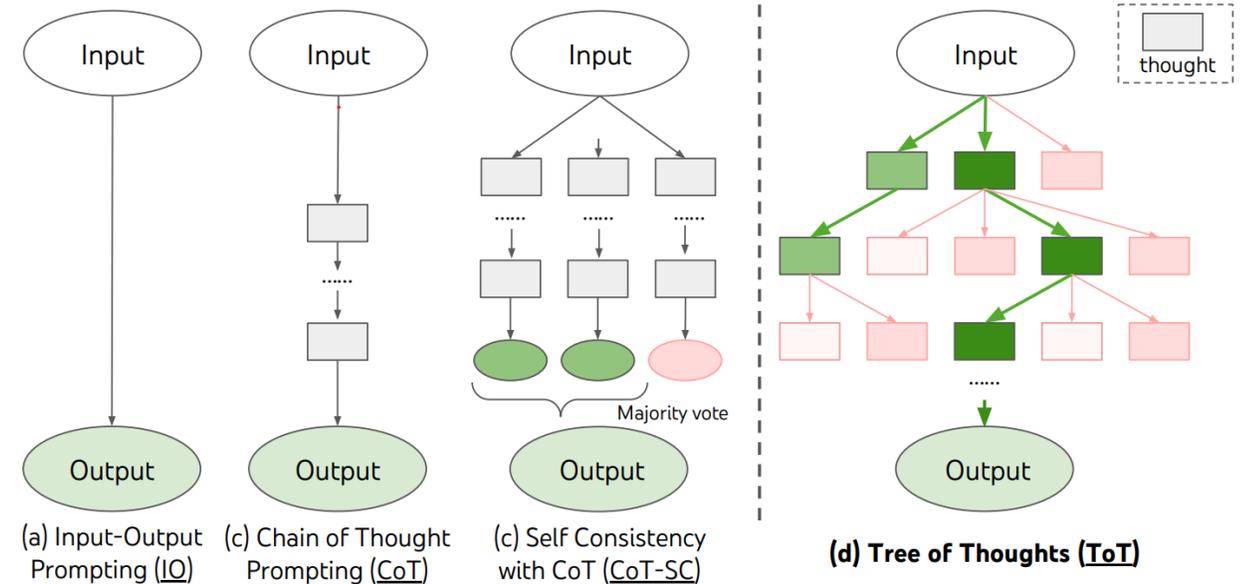
Deliberate Problem Solving with Large Language Models

Erlaubt LLMs Denk- und Lösungswege zu erkunden.
Problem wird als Baumstruktur modelliert.

- 1. Zerlegung von Gedanken**
Zwischenprozesse werden in einzelne Denkschritte zerlegt
- 2. Gedanken Generator**
Strategien werden vorgeschlagen, um Kandidaten für den nächsten Denkschritt zu generieren
- 3. Bewertung von Zuständen**
Zustandsbewerter bewerten den Fortschritt in Bezug auf die Problemlösung
- 4. Suchalgorithmus (nach den erfolgversprechendsten Gedanken)**
Unterschiedliche Suchalgorithmen, je nach Struktur des Baumes

<https://arxiv.org/abs/2305.08291>

Shunyu Yao, Princeton University
Dian Yu, Google DeepMind
Jeffrey Zhao, Google DeepMind
Izhak Shafran, Google DeepMind
Thomas L. Griffiths, Princeton University
Yuan Cao, Google DeepMind
Karthik Narasimhan, Princeton University



ToT ist ein Rahmenwerk, das Sprachmodelle dazu befähigt, autonomer und intelligenter Entscheidungen zu treffen und Probleme zu lösen.

Es gibt nicht die eine Strategie ...

- *Es können auch verschiedene Strategien kombiniert werden*
- *Z.B. im Rahmen eines "Chain of Thought", Teile des "Tree of Thought" durch das LLM selbst simulieren lassen:*

Imagine three different experts are answering this question.

All experts will write down 1 step of their thinking, then share it with the group.

Then all experts will go on to the next step, etc.

If any expert realizes they're wrong at any point, then they leave.

The question is...

Es ist wichtig, für spezifische Anwendungsfälle das Modell zu wählen und Kosten und Strategien zu erarbeiten.

Was bedeutet das für zukünftige Unternehmenssoftware?

- **Benutzer werden es in Zukunft erwarten!**
- *Es eröffnet möglicherweise neue und effizientere Abläufe*
- *Schulungsaufwände können weiter sinken*
- *Zum Zeitpunkt der Erstellung von Software **unbekannte** Anforderungen können abgebildet werden*
- *Durch Angebote wie "**Microsoft Azure OpenAI Services**" oder auch die Nutzung von Offline Open Source Modellen wie "**Meta's Llama 2**" kann auch der Datenschutz gewahrt und ggf. Kosten begrenzt werden.*
- *Open Source Frameworks wie MS Semantic Kernel oder LangChain vereinfachen die Interaktion mit LLMs*



Es macht Sinn, diese Möglichkeiten zu evaluieren!

Wie moderne KI die klassische Softwareentwicklung transformiert



Demo Time

Demo Time

- „Copilot“ Beispielanwendung
- Semantische Funktionen und Zugriff auf Daten
- Custom Planner und AI Plugins



Christian Kratky
Principal Software Engineer
prodot GmbH

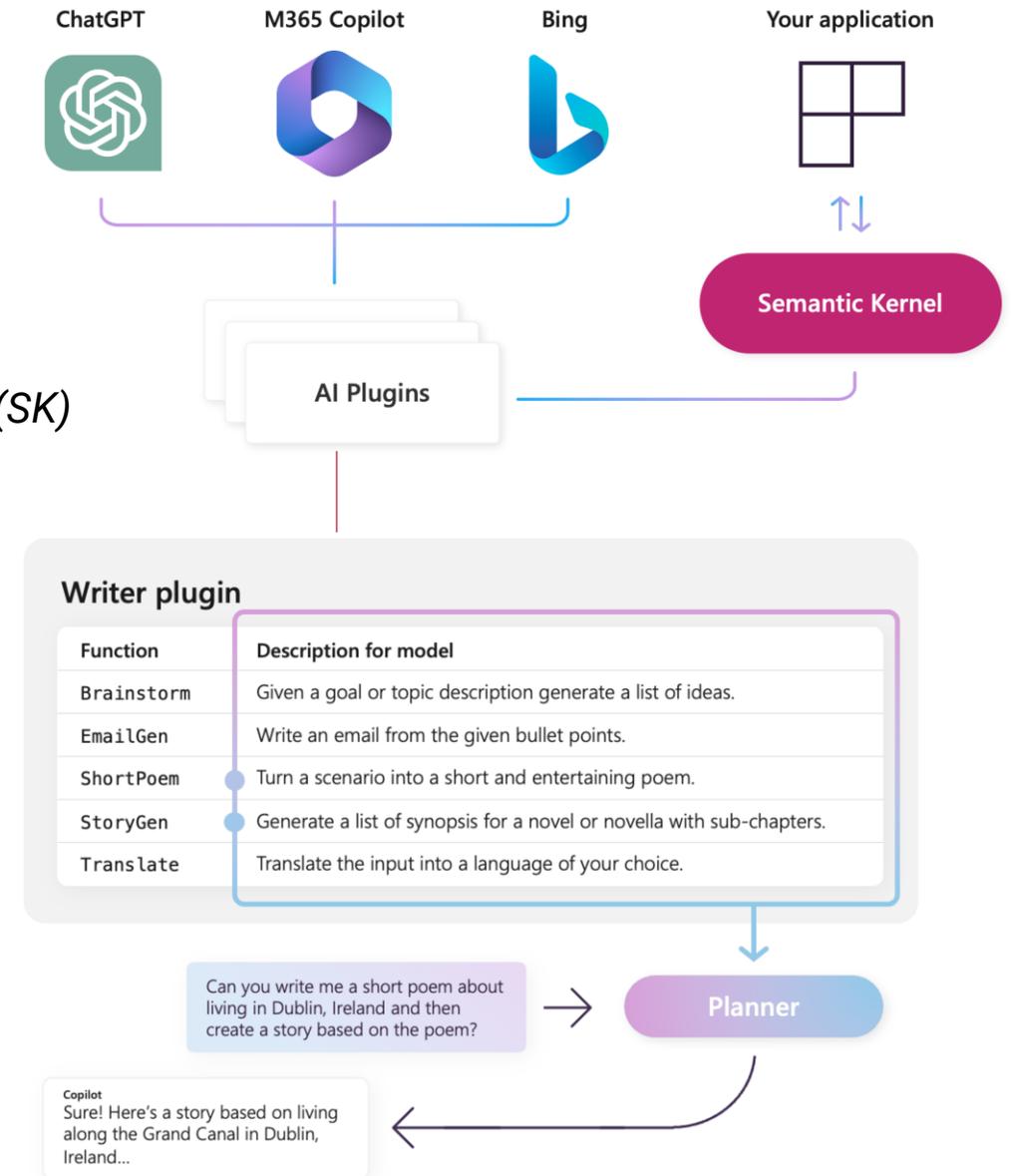


Stefan Wilkes
Unit Lead / Senior Software Architect
prodot GmbH

AI Plugins

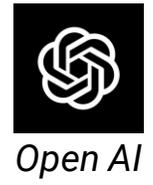
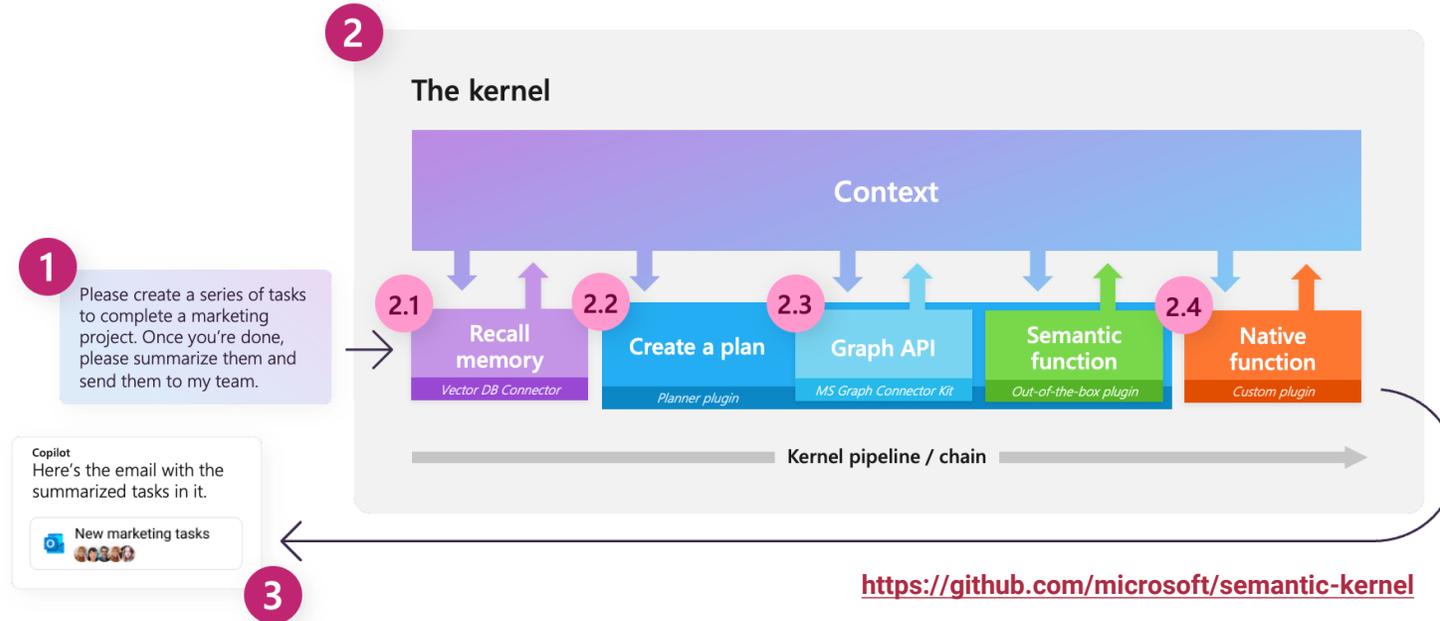
- *Plugins kapseln KI-Funktionen in einzelne Funktionseinheiten.*
- *Plugins oder auch Skills sind die Bausteine des Semantic Kernel (SK) und - dank der OpenAI Plugin Spezifikation - auch anderer Plattformen und Orchestratoren*
- *Plugins bestehen aus Funktionen die den AI-Apps & -Services bekannt gemacht werden*
 - *Semantische Funktionen: Funktionen die das Modell anweisen wie eine Eingabe zu interpretieren ist*
 - *Native Funktionen: Funktionen die eine Eingabe des Nutzers oder des Modells „lokal“ auswerten und verarbeiten -> Klassischer Code*
- *Funktionen werden entweder verkettet (Chaining) oder automatisch von einem Planer ausgeführt*

Viel wichtiger ist aber nicht die Funktion an sich sondern deren semantische Beschreibung!



Semantic Kernel (SK)

Der Kernel steuert die Anfrage durch die Ausführung einer zuvor definierten Pipeline



...

- Weitere Komponenten des Semantic Kernel

- Memories / Embeddings / Vektor Datenbank (2.1)
- Planner (2.2)
- Konnektoren zu Graph API, Azure Services, ... (2.3)
- Standardskills wie ConversationSummarySkill, MathSkill, HttpSkill, TextSkill, TimeSkill, ... (2.4)



Planner

- *Planner des Semantik Kernel*
 - *Sequential Planner (C#)*
 - *Basic Planner (C#, Python)*
 - *Action Planner (C#)*
- *Mögliche Problematiken*
 - *Aktuell nur lineare Planung*
 - *Bedingungen sind schwer abzubilden*
 - *Geringe Anpassbarkeit und intransparente Token Nutzung*
 - *Halluzinieren durch nicht notwendige Funktionen*
- *Mögliche Lösung*
 - *Custom Planner, denn Planner sind auch nur semantische und native Funktionen*

Der Planer verwendet unsere Funktionen auf eine Weise, an die wir vielleicht während der Entwicklung noch gar nicht gedacht haben

```
1 You are a planner for the Semantic Kernel.
2 Your job is to create a properly formatted JSON plan step by step, to satisfy the goal given.
3 Create a list of subtasks based off the [GOAL] provided.
4 Each subtask must be from within the [AVAILABLE FUNCTIONS] list. Do not use any functions that are not in the list.
5 Base your decisions on which functions to use from the description and the name of the function.
6 Sometimes, a function may take arguments. Provide them if necessary.
7 The plan should be as short as possible.
8 For example:
9
10 [AVAILABLE FUNCTIONS]
11 EmailConnector.LookupContactEmail
12 description: looks up the a contact and retrieves their email address
13 args:
14 - name: the name to look up
15
16 WriterSkill.Translate
17 description: translate the input to another language
18 args:
19 - input: the text to translate
20 - language: the language to translate to
21
22 WriterSkill.Summarize
23 description: summarize input text
24 args:
25 - input: the text to summarize
26
27 FunSkill.Joke
28 description: Generate a funny joke
29 args:
30 - input: the input to generate a joke about
31
32 [GOAL]
33 "Tell a joke about cars. Translate it to Spanish"
34
35 [OUTPUT]
36 {
37   "input": "cars",
38   "subtasks": [
39     {"function": "FunSkill.Joke"},
40     {"function": "WriterSkill.Translate", "args": {"language": "Spanish"}}
41   ]
42 }
43
44 ...
45
46 [AVAILABLE FUNCTIONS]
47 {{$available_functions}}
48
49 [GOAL]
50 {{$goal}}
51
52 [OUTPUT]
```

Kosten

- Kurzer Exkurs zu Token
 - Bei der Tokenisierung werden die **Eingabe- und Ausgabertexte** in kleinere Einheiten (Wörter, Subwörter, Symbole) zerlegt
 - OpenAI und Azure OpenAI nutzen Byte-Pair-Encoding (BPE) wo häufig vorkommende Zeichenfolgen zu einem einzigen Token zusammen gefasst werden.
- Kosten / 1k Tokens *Stand 23.08.2023*
 - GPT3.5 (4K): In \$0.0015, Out \$0.002
 - GPT4 (8K): In \$0.03, Out \$0.06
- Azure OpenAI in mehr und mehr Regionen verfügbar
- Kostenreduktion durch
 - Prompt-Optimierung
 - Vorgefertigte Pläne (Bake, don't fry)
 - Optimierte Modelle
 - Embeddings

Vielleicht ein Thema für's nächste Meetup ;)

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🍌

Sequences of characters commonly found next to each other may be grouped together: 1234567890

Clear

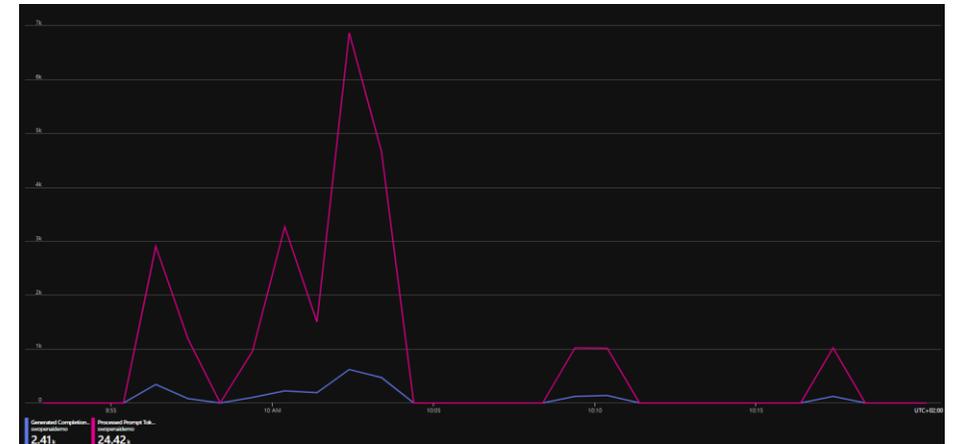
Show example

Tokens	Characters
64	252

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🍌

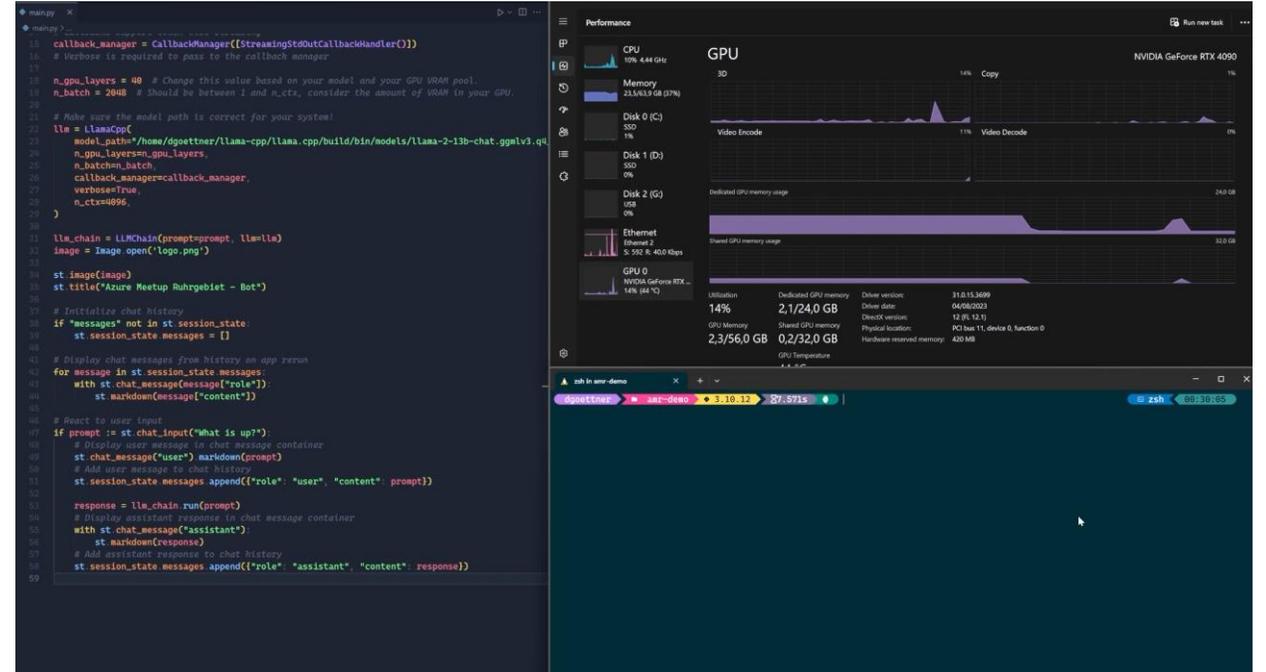
Sequences of characters commonly found next to each other may be grouped together: 1234567890



Ausblick: Self-Hosting

Metas Open Source Model LLaMa 2

- **Pro**
 - Open Source (Kommerzielle Nutzung erlaubt)
 - Offline Nutzung
 - Kosten? (Siehe Performance)
 - Datenschutz
- **Contra**
 - Performance? (Siehe Kosten)
- **Laut Meta schneidet LLaMa (13B) teilweise besser ab als GPT3.5 (175B) und ist dabei signifikant schlanker ***



* [LLaMA: Open and Efficient Foundation Language Models - Meta Research \(facebook.com\)](https://arxiv.org/abs/2307.08637)



```
File Edit Selection View Go Run ...
main.py x
main.py >
15 callback_manager = CallbackManager([StreamingStdOutCallbackHandler()])
16 # Verbose is required to pass to the callback manager
17
18 n_gpu_layers = 40 # Change this value based on your model and your GPU VRAM pool.
19 n_batch = 2048 # Should be between 1 and n_ctx, consider the amount of VRAM in your GPU.
20
21 # Make sure the model path is correct for your system!
22 llm = LlamaCpp(
23     model_path="/home/dgoettner/llama-cpp/llama.cpp/build/bin/models/llama-2-13b-chat.ggmlv3.q4
24     n_gpu_layers=n_gpu_layers,
25     n_batch=n_batch,
26     callback_manager=callback_manager,
27     verbose=True,
28     n_ctx=4096,
29 )
30
31 llm_chain = LLMChain(prompt=prompt, llm=llm)
32 image = Image.open('logo.png')
33
34 st.image(image)
35 st.title("Azure Meetup Ruhrgebiet - Bot")
36
37 # Initialize chat history
38 if "messages" not in st.session_state:
39     st.session_state.messages = []
40
41 # Display chat messages from history on app rerun
42 for message in st.session_state.messages:
43     with st.chat_message(message["role"]):
44         st.markdown(message["content"])
45
46 # React to user input
47 if prompt := st.chat_input("What is up?"):
48     # Display user message in chat message container
49     st.chat_message("user").markdown(prompt)
50     # Add user message to chat history
51     st.session_state.messages.append({"role": "user", "content": prompt})
52
53     response = llm_chain.run(prompt)
54     # Display assistant response in chat message container
55     with st.chat_message("assistant"):
56         st.markdown(response)
57     # Add assistant response to chat history
58     st.session_state.messages.append({"role": "assistant", "content": response})
59
```

Task Manager

Performance

CPU: 6% 4.30 GHz

Memory: 24.5/63.9 GB (38%)

Disk 0 (C:): SSD 0%

Disk 1 (D:): SSD 0%

Disk 2 (G:): USB 0%

Ethernet: Ethernet 2 S: 16.0 R: 0 Kbps

GPU 0: NVIDIA GeForce RTX ... 3% (43 °C)

GPU

NVIDIA GeForce RTX 4090

3D: 0% Copy: 0%

Video Encode: 0% Video Decode: 0%

Dedicated GPU memory usage: 24.0 GB

Shared GPU memory usage: 32.0 GB

Utilization	Dedicated GPU memory	Driver version:	31.0.15.3699
3%	1,9/24,0 GB	Driver date:	04/08/2023
GPU Memory	Shared GPU memory	DirectX version:	12 (FL 12.1)
2,0/56,0 GB	0,1/32,0 GB	Physical location:	PCI bus 11, device 0, function 0
GPU Temperature	GPU Temperature	Hardware reserved memory:	420 MB
43 °C	43 °C		

zsh in amr-demo

dgoettner amr-demo 3.10.12 87.571s

zsh 00:30:05

WSL: Ubuntu-20.04

Ln 59, Col 1 Spaces: 4 UTF-8 LF Python 3.10.12 64-bit Prettier

Wie moderne KI die klassische Softwareentwicklung transformiert



Wrap Up

Wrap-up

- *„Mindset Shift“*
- *Generative KI*
 - *„Geht nicht mehr weg“ und wird in Software zukünftig erwartet werden*
 - *Zitat: „In no more than 3 years ... broken or invisible!“*
 - *eröffnet komplett neue Möglichkeiten*
 - *einige Ansätze haben wir vorgestellt*
- *Auch Experimentieren lohnt sich, um Erfahrungen für heutige oder zukünftige Szenarien zu sammeln.*

Wir unterstützen gerne beim Brainstorming oder konkreten Ideen.



prodot - KI Headstart Workshop

- *Aktuelle KI-Entwicklungen in der Übersicht*
- *Kennenlernen des Geschäftsmodells und der Prozesse*
- *Identifizieren sinnvoller Einsatzszenarien für generative KI*
- *Unternehmens-Governance und Compliance-Vorgaben*
- *Ableiten von Handlungsschwerpunkten und einer Roadmap*



Ideen reflektieren

Datenschutzthemen

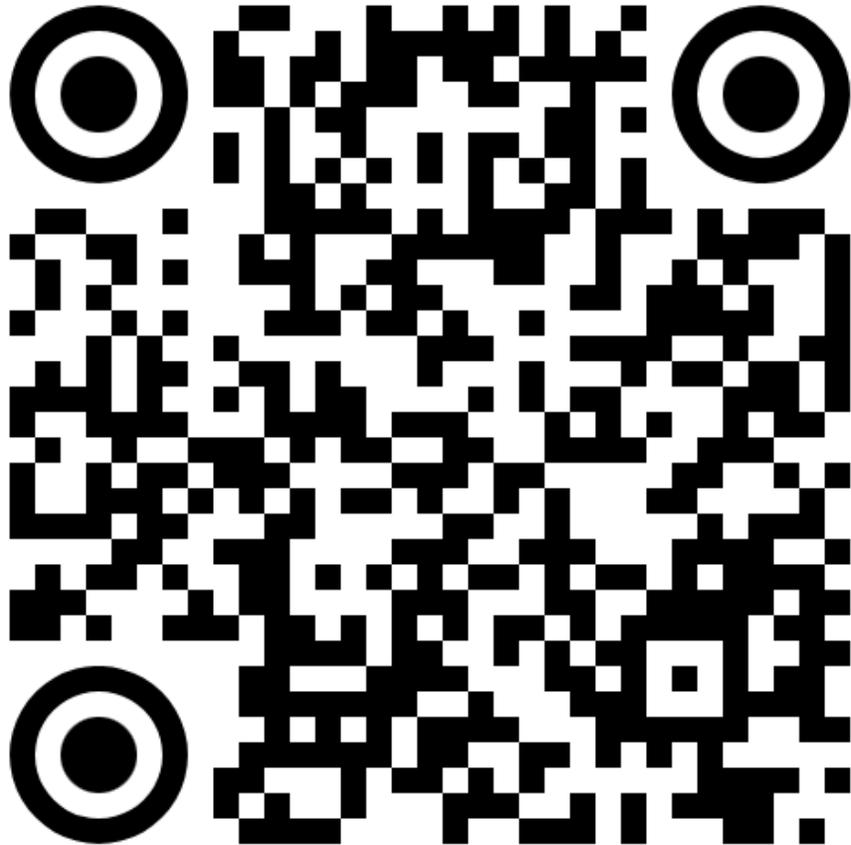
Use Case Brainstorming

Experimente

Proof of Concepts

Prototyping

prodot - KI Headstart Workshop



Ideen reflektieren

Datenschutzthemen

Use Case Brainstorming

Experimente

Proof of Concepts

Prototyping



Vielen Dank



Mischa Hüschen
Managing Director
prodot GmbH

mischa.hueschen@prodot.de



Christian Kratky
Principal Software Engineer
prodot GmbH

christian.kratky@prodot.de



Stefan Wilkes
Unit Lead / Senior Software Architect
prodot GmbH

stefan.wilkes@prodot.de