

Navigating Tomorrow – Die Evolution der IT-Arbeits- welt im Aufkommen der KI

Autoren: Vorstand des AK Software Engineering & Software Architektur

Die Zukunft der IT-Arbeitswelt steht bevor und so luden wir die Mitglieder des Arbeitskreises ein, Teil dieses aufregenden Wandels zu sein! Unsere Arbeitskreis-Sitzung war ein Sprungbrett für Ideen, Innovationen und Diskussionen zur Evolution der IT-Arbeitswelt im Zeitalter der KI.

Unter dem Titel »Navigating Tomorrow – Die Evolution der IT-Arbeitswelt im Aufkommen der KI« fand am 20.04.2024 eine inspirierende Sitzung mit Vordenkern, Experten und Visionären auf dem Gebiet statt.

Inmitten faszinierender Diskussionen zu Themen wie KI-Unterstützung für heutige Software Engineering Rollen, dem Aufkommen neuer Jobs, der ethischen Verantwortung in der KI-Nutzung und vielen weiteren spannenden Facetten dieses neuen Zeitalters wagen wir einen Blick in die Zukunft, wie diese gestaltet werden kann.

Die Diskussionsschwerpunkte der Arbeitskreissitzung wurden durch die folgenden Präsentationen und Impulse gesetzt:

Schwerpunkt 1: KI-Unterstützung für heutige Software Engineering Rollen: »Automatisierung« von kreativen Prozessen.

Prof. Dr.-Ing Sabine Radomski stellte in Ihrem Vortrag anhand von drei Beispielen vor, wie Fuzzy-Logik an unterschiedlichen Stellen im Prozess der Softwareentwicklung unterstützen kann.

- Im Requirements Engineering lassen sich Anforderungen mit KI (Fuzzy-Regeln) anhand definierter Qualitätskriterien, z. B. nach ISO-Standards bewerten.
- Im Projektmanagement lässt sich die Auswahl eines geeigneten Vorgehensmodells über KI (Fuzzy Regeln) unterstützen, wenn geeigneter Kriterien definiert wurden.
- Im Software Engineering kann die Auswahl einer passenden Software-Architektur mittels KI unterstützt werden, wenn geeignete Fuzzy Kriterien definiert wurden.

Key Takeaways

- KI ist mehr als generative KI und Chat-GPT.
- Fuzzy Logic ist eine sehr gute Methode um mit KI zu arbeiten: Änderungen sind sofort sichtbar und es gibt transparente Ergebnisse.

KI ist mehr als generative
KI und Chat-GPT.

- Im Vergleich zu neuronalen Netzen, benötigen Fuzzy basierte Ansätze weniger Ressourcen & Daten um lauffähig gemacht und genutzt zu werden.
- Am Ende gilt auch bei der Fuzzy Automatisierung: Der Mensch entscheidet auf Grundlage seiner Erfahrung, ob er die Ergebnisse der KI verwendet oder nicht.
- Die moralische Entscheidung, ob Ergebnisse verwendet werden, liegt weiterhin beim Menschen.
- Die Haftbarkeitsfrage für die Verwendung von KI Ergebnissen liegt bei demjenigen der diese Ergebnisse verwendet.

Der Mensch entscheidet auf Grundlage seiner Erfahrung, ob er die Ergebnisse der KI verwendet oder nicht.

Schwerpunkt2: Künstliche Intelligenz – das neue Multi-Tool!?

Dinko Eichin Roitman präsentierte in seinem Vortrag konkrete Fälle, in denen KI durch effektive Ergänzung und Unterstützung menschlicher Tätigkeiten erheblichen Mehrwert schafft und KI zu einem unverzichtbaren Werkzeug geworden ist.

- In der Softwareentwicklung hilft KI durch Verbesserung der Entscheidungsfindung über Assistenzfunktionen wie z. B. in Copilot und anderen SWE-Tools.
- In der personalisierten Medizin unterstützt KI an den entlegensten Stellen der Welt Ärzte und Ärztinnen und ist so bereits zu einem unverzichtbaren Werkzeug geworden.

Key Takeaways

- Problem: Ohne weiteres Hintergrundwissen kann KI den Menschen Angst machen.
- Es ist wichtig positive Beispiele für KI Einsatz zu sammeln und zu verbreiten um neben den Risiken auch die Chancen transparent zu machen.
- KI kann als Chance für Chancengleichheit im Gesundheitswesen gesehen werden, da sie überall auf der Welt verfügbar ist. Es gibt bereits heute eine Reihe sehr sinnvoller Anwendungen: Anamnese durch Augenscan, medizinische Prothesen, etc.
- Im Software Engineering kann KI verwendet werden um automatisch eine technische Dokumentation für die spätere Weiterentwicklung und Pflege zu erstellen.
- Interessante KI Services im Software Engineering: Eightify, html.to.design, Igniter.ai, runway, scholarcy, whisper Memos, midjourney, Learn Prompting, DeepLearning.ai, Elements Of AI, Haystack, LlamaIndex, Python, HuggingFace, AlephAlpha, OpenAI.
- Um niemanden auf dem Weg zu verlieren kann es sich lohnen für alle Mitarbeitenden eine kleine »Grundausbildung« anzubieten, die Ängste nimmt und die Neugier steigert.
- Wichtig ist das Verständnis, dass Prompt-Engineering eine iterative Arbeit ist, bei der sich Mensch und Maschine schrittweise nähern um bessere Ergebnisse zu erzielen.

Es ist wichtig positive Beispiele für KI Einsatz zu sammeln und zu verbreiten um neben den Risiken auch die Chancen transparent zu machen.

Schwerpunkt 3: Tab-Engineering – AI Code Completion für eine bessere Developer Experience

Torben Köhn erläuterte in seiner Live-Hacking Session Tab-Engineering am Beispiel von Copilot.

Key Takeaways

- Neue Disziplin im Software Engineering: Tab Engineering beschleunigt Entwicklung.
- Copilot erkennt den Code Stil.
- Copilot erlaubt Re-factoring per Inlinechat.
- Copilot lernt mit – Vorschläge werden im Laufe der Zeit immer besser.
- Tools wie Copilot et al erhöhen die Sprintgeschwindigkeit signifikant. Aus eigener Erfahrung kann eine Effektivitätssteigerung zwischen 30 – 60 Prozent erzielt werden.
- Thema Datenschutz: der eigene Code wird – wenn nicht explizit ausgeschlossen – geteilt. Hierbei ist wichtig zu verstehen welche Bestandteile sicherheitskritisch oder businesskritisch sind! Das Gute daran – oft ist weniger kritisch als auf den ersten Blick angenommen.
- Vorsicht ist allerdings bei der Copyrightsituation geboten- Copilot wird mit OSS trainiert. Copyrightsituation ist nicht eindeutig da unterschiedliche Quellen und Lizenzmodelle zusammenkommen.

Neue Disziplin im Software Engineering: Tab Engineering beschleunigt Entwicklung.

Schwerpunkt 4: Wie AI sich selbst im Software Engineering abschafft.

Martin Fix stellte in seinem Lightning Talk die provokative These auf, dass sich KI selbst auf lange Zeit abschafft – oder nicht!?

Durch den Einsatz von KI in der Softwareentwicklung, Beispiel Copilot, werden immer weniger Junior-Softwareentwickler benötigt, da KI – gut bedient – bereits einen guten ersten Job macht, der sicherlich dem einen oder anderen Junior Softwareentwickler ebenbürtig ist. Es braucht also nur noch Senior Softwareentwickler die am Ende das Ergebnis der KI kontrollieren und Feintunen. Senioreentwickler fangen aber als Junioreentwickler an, die es dann nicht mehr gibt. Woher kommt die Erfahrung, die die KI trainiert?

Soweit die Hypothese. Sterben uns also mittelfristig alle Senior Softwareentwickler aus, weil wir den Nachwuchs durch KI ersetzt haben? Überprüft die KI sich zukünftig selbst? Wer trägt dann die Verantwortung? Was wird aus Copyright – was aus Intellectual Properties?

Key Takeaways

- Es ist stand heute unwahrscheinlich, dass KI Dinge inkl. Software selbstständig entwirft, konzipiert, entwickelt, testet und betreibt – aber wie sieht es in 50 Jahren aus?
- KI Tests können heute bereits KI Code testen. Beispiel: Test Fuzzing von APIs.
- Wir müssen jetzt damit anfangen uns Gedanken zu machen, wie wir KI einsetzen wollen und wie zukünftige Rollen und Berufe im Software Engineering dadurch beeinflusst werden.

Wir müssen jetzt damit anfangen uns Gedanken zu machen, wie wir KI einsetzen wollen.

- Letztendlich benötigen wir eine taktische Verantwortung (Compliance, Prüfbarkeit) und eine strategische Verantwortung (Richtlinien, Vorschriften).
- Technikfolgen werden in der Zukunft versicherbar sein müssen, wie beispielsweise beim autonomen Fahren durch die Haftpflicht- oder Kaskoversicherung.

Fazit

KI ist eine weltverändernde Technologie, die nahezu überall verfügbar und aus diesem Grunde schwer kontrollierbar ist. Wie und wofür genau KI zukünftig überall eingesetzt wird ist kaum vorauszusagen. Allerdings wird uns KI in Zukunft in vielen Bereichen unterstützen – egal ob im Software Engineering oder in der Pflege. Auch wenn wir stets an das Gute im Menschen glauben, müssen wir uns schon heute Gedanken darüber machen, wie und wo KI sinnvoll eingesetzt werden kann und wo es reguliert werden muss ohne Chancen und Fortschritt zu verhindern.



Dr. Frank Termer
Bereichsleiter Software
T 030 27576-232
f.termer@bitkom.org