

Leistungsfähigkeit von Microsoft Copilot bei der Automatisierung von Microsoft Power Automate Workflows im Vergleich zur manuellen Erstellung

Bachelorarbeit

eingereicht von: **Ing. Johannes Lienbacher**
Matrikelnummer: 52108400

im Fachhochschul-Bachelorstudiengang Wirtschaftsinformatik (0470)
der Ferdinand Porsche FernFH

zur Erlangung des akademischen Grades eines
Bachelor of Arts in Business

Betreuung und Beurteilung: Prof. Dr. Joachim Steinwendner, MSc

St.Georgen an der Leys, Mai 2025

Ehrenwörtliche Erklärung

Ich versichere hiermit,

1. dass ich die vorliegende Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Inhalte, die direkt oder indirekt aus fremden Quellen entnommen sind, sind durch entsprechende Quellenangaben gekennzeichnet.
2. dass ich diese Bachelorarbeit bisher weder im Inland noch im Ausland in irgendeiner Form als Prüfungsarbeit zur Beurteilung vorgelegt oder veröffentlicht habe.

St.Georgen an der Leys, 24.05.2025


Unterschrift

Creative Commons Lizenz

Das Urheberrecht der vorliegenden Arbeit liegt bei Johannes Lienbacher. Sofern nicht anders angegeben, sind die Inhalte unter einer Creative Commons <„Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz“ (CC BY-NC-SA 4.0)> lizenziert.

Die Rechte an zitierten Abbildungen liegen bei den in der jeweiligen Quellenangabe genannten Urheber*innen.

Die Kapitel 2 bis 3 der vorliegenden Bachelorarbeit wurden im Rahmen der Lehrveranstaltung „Bachelor Seminar 1“ eingereicht und am 31.01.2025 als Bachelorarbeit 1 angenommen.

Kurzzusammenfassung: Leistungsfähigkeit von Microsoft Copilot bei der Automatisierung von Microsoft Power Automate Workflows im Vergleich zur manuellen Erstellung

Automatisierungsprozesse sind entscheidend für die Produktivität und Effizienz moderner Unternehmen. Diese Arbeit untersucht, ob Microsoft Copilot, ein KI-gestütztes Tool innerhalb von Microsoft Power Automate, Workflows effizienter und schneller erstellt als manuelle Methoden, und dabei weniger fehleranfällig ist. In einer experimentellen Fallstudie wurden fünf praxisnahe Szenarien sowohl manuell als auch KI-unterstützt umgesetzt und anhand der Kriterien Durchlaufzeit, Fehleranfälligkeit und zusätzlich Erstellungsdauer bewertet. Die Ergebnisse zeigen, dass Copilot-generierte Workflows zwar teilweise schneller ausgeführt werden, jedoch bei der Erstellung deutlich zeitintensiver und fehleranfälliger sind. Damit ist Copilot aktuell noch keine vollständige Alternative für komplexe Workflows, könnte aber bei einfachen Automatisierungen und für erfahrene Anwender unterstützend wirken.

Schlagwörter:

Microsoft Copilot, Microsoft Power Automate, Workflow-Automatisierung, KI-gestützte Prozessautomatisierung, Low-Code-Plattform, Citizen Development, Effizienz von Workflows

Abstract: Performance of Microsoft Copilot in Automating Microsoft Power Automate Workflows Compared to Manual Creation

Automation processes are essential for the productivity and efficiency of modern enterprises. This thesis investigates whether Microsoft Copilot, an AI-based tool integrated within Microsoft Power Automate, creates workflows more efficiently, faster, and with fewer errors than manual methods. Through an experimental case study, five practical scenarios were implemented both manually and AI-assisted, evaluating workflow execution time, error susceptibility and additionally creation duration. Results reveal that workflows generated by Copilot run partially faster operationally but take significantly longer to create and are more prone to errors. Thus, Copilot currently does not fully substitute manual creation for complex workflows but may serve as a supportive tool for simpler automations and experienced users.

Keywords:

Microsoft Copilot, Microsoft Power Automate, Workflow automation, AI-assisted process automation, Low-code platform, Citizen Development, Workflow efficiency

Inhaltsverzeichnis

1. EINLEITUNG	1
1.1 Problemstellung	1
1.2 Forschungsfrage	1
1.3 Aufbau der Arbeit	2
1.4 Theoretischer Rahmen	2
2. THEORETISCHE GRUNDLAGEN	4
2.1 Grundlagen von Microsoft Copilot für M365	4
2.1.1 Einführung in den Microsoft Copilot für M365	4
2.1.2 Technische Architektur und Funktionsweise	4
2.1.3 Grundlagen zu Large Language Models (LLMs)	7
2.1.4 Prompt Engineering	8
2.1.5 Sicherheit und Datenschutzaspekte	9
2.2 Grundlagen Microsoft Power Automate	9
2.2.1 Einführung in Microsoft Power Automate	10
2.2.2 Funktionsweise von Microsoft Power Automate	10
2.2.3 Low-Code	11
2.2.4 Grundlagen zu Robot Process Automation (RPA)	11
2.3 Grundlagen der KI-gestützten Workflow Automatisierung	11
2.4 Effizienz und Fehleranfälligkeit bei der KI-gestützten Programmierung	12
2.4.1 Effizienz durch KI-gestützte Automatisierung	12
2.4.2 Herausforderungen durch Fehleranfälligkeit	13
2.4.3 Daten zur Effizienz und Validierung	13
2.5 Literaturanalyse	14
2.5.1 Die Rolle von LLMs in der Automatisierung	14
2.5.2 Innovationen in der RPA durch KI	14
2.5.3 Effizienz und Fehleranfälligkeit von KI-gestützten Workflows	14
2.5.4 Relevanz für die Bachelorarbeit	15
3. METHODIK	16

3.1 Auswahl der Use-Cases	16
3.2 Experimentelle Fallstudie	17
3.2.1 Die Fallstudie	17
3.2.2 Das Experiment	18
3.2.3 Die Experimentelle Fallstudie	18
3.3 Bewertungskriterien	18
3.3.1 Fehleranfälligkeit	18
3.3.2 Schnelligkeit (Durchlaufzeit)	19
3.3.3 Effizienz	19
4. DURCHFÜHRUNG DER FALLSTUDIE	21
4.1 Praktisches Beispiel: Erstellung eines Workflows mit Microsoft Copilot	21
4.1.1 Ausgangslage und Ziel	21
4.1.2 Ausgangs Prompt (Use-Case Definition)	21
4.1.3 Finaler Prompt	23
4.1.4 Anpassung mit Copilot im Flow-Builder	26
4.2 Use-Case 1: Benachrichtigung bei Datei-Upload in SharePoint	27
4.3 Use-Case 2: Extraktion von Tasks aus eingehenden E-Mails	28
4.4 Use-Case 3: Zusammenfassung neuer SharePoint-News per E-Mail	29
4.5 Use-Case 4: Automatisierte Genehmigung für Zugriff auf Teams-Team	30
4.6 Use-Case 5: Automatische Teams-Zugriffsberechtigung für neue IT-Mitarbeiter	30
5. DISKUSSION UND BEANTWORTUNG DER FORSCHUNGSFRAGE	32
5.1 Zusammenfassung der Ergebnisse	32
5.2 Beantwortung der Forschungsfrage im Kontext der Hypothese	33
5.2.1 Effizienz (Dauer der Workflow-Erstellung)	34
5.2.2 Schnelligkeit (Durchlaufzeit der Workflows)	34
5.2.3 Fehleranfälligkeit	34
5.2.4 Beantwortung der Forschungsfrage	35
5.3 Vergleich mit der Literatur	35
5.3.1 Übereinstimmungen mit bestehenden Erkenntnissen	35
5.3.2 Unterschiede und Ergänzungen zur Literatur	36

5.3.3 Implikationen aus dem Literaturvergleich	36
6. SCHLUSSFOLGERUNG UND AUSBLICK	37
6.1 Schlussbetrachtung der gesamten Bachelorarbeit	37
6.2 Fazit aus dem empirischen Teil	37
6.3 Implikation für die Praxis und zukünftige Forschung	37

1. Einleitung

1.1 Problemstellung

Automatisierungsprozesse (Workflows) sind ein entscheidender Faktor für die Effizienz moderner Geschäftsprozesse und bieten Unternehmen die Möglichkeit, operative Aufgaben nachhaltig zu optimieren. Die Plattform Microsoft Power Automate stellt dabei eine Schlüsseltechnologie dar, die insbesondere durch ihren Low-Code-Ansatz den sogenannten „Citizen Developern“ ermöglicht, eigenständig und ohne tiefgehende Programmierkenntnisse Prozesse zu automatisieren. Die Einführung des KI-gestützten Tools Microsoft Copilot für die Microsoft 365-Umgebung (M365) verspricht dabei eine zusätzliche Vereinfachung der Workflow-Erstellung durch die Verwendung natürlicher Sprache. Erste praktische Anwendungen zeigen jedoch, dass diese Vereinfachung derzeit nur bedingt gegeben ist und oft erhebliche manuelle Nachkorrekturen notwendig sind. Insbesondere bei komplexeren Anforderungen stößt Copilot an seine Grenzen, was die erhofften Vorteile für weniger erfahrene Anwender relativiert. Microsoft Copilot für M365 kann innerhalb von Microsoft Power Automate eingesetzt werden, um durch sprachliche Eingaben (Prompts) Automatisierungsprozesse zu erstellen (Hyun et al., 2021).

Da der spezifische Vergleich von manueller Erstellung von Workflows im Vergleich zur automatisierten Erstellung von Workflows in der Forschung noch wenig thematisiert wurde, soll diese Arbeit Aufschluss darüber geben. Zur Beantwortung dieser Frage wird im Rahmen einer experimentellen Fallstudie untersucht, ob die von Microsoft Copilot für M365 erstellten Microsoft Power Automate Workflows manuell erstellten Microsoft Power Automate Workflows überlegen sind (Hyun et al., 2021).

1.2 Forschungsfrage

Das Ziel dieser Bachelorarbeit ist es herauszufinden, ob die Erstellung von Microsoft Power Automate Workflows mit Microsoft Copilot für M365 im Vergleich zu manuell erstellten Workflows effizienter, schneller (Durchlaufzeit) und mit einer geringeren Fehleranfälligkeit funktioniert. Diese Fragestellung ist spannend, da sie Aufschluss über die Leistungsfähigkeit von Microsoft Copilot für M365 im Zusammenspiel mit Microsoft Power Automate gibt und zugleich die Rolle von IT-Grundwissen in der Workflow-Erstellung beleuchtet.

Meine Hypothese zu diesem Thema ist, dass die von Microsoft Copilot erstellten Microsoft Power Automate Workflows effizienter und schneller sind, jedoch eine höhere Fehleranfälligkeit im Vergleich zu manuell erstellten Workflows aufweisen.

1.3 Aufbau der Arbeit

Der Aufbau dieser Bachelorarbeit ist darauf ausgelegt, die Forschungsfrage systematisch zu beantworten und eine fundierte Bewertung der Effizienz, Fehleranfälligkeit und Schnelligkeit von KI-gestützten und manuell erstellten Workflows zu ermöglichen. Nach der Einleitung, in der in das Thema eingeführt, die Problemstellung erläutert und die Zielsetzung der Arbeit beschrieben wird, widmet sich das zweite Kapitel den theoretischen Grundlagen. Hier werden die relevanten Technologien wie Microsoft Copilot, Microsoft Power Automate und Large Language Models (LLMs) sowie die Mechanismen der Workflow-Automatisierung erklärt. Dieses Kapitel bildet das Fundament, auf dem die experimentelle Fallstudie aufbaut.

Das Kapitel „Methodik“ beschreibt das Vorgehen der Untersuchung. Zunächst werden die Use-Cases vorgestellt, die als Basis für die Vergleichsanalyse dienen. Darauf folgt die Erläuterung des experimentellen Ansatzes, der eine Kombination aus Fallstudie und Experiment umfasst, um sowohl praxisnahe als auch kontrollierte Bedingungen zu berücksichtigen. Es werden die angewandten Messmethoden detailliert erklärt, mit denen die Daten für die Bewertung der Workflows erhoben werden. Schließlich werden die Bewertungskriterien dargelegt, anhand derer die Ergebnisse interpretiert und miteinander verglichen werden.

Im Kapitel „Durchführung der Fallstudie“ werden die Ergebnisse der Fallstudie präsentiert. Die Datenanalyse fokussiert sich auf die Effizienz, Fehleranfälligkeit und Schnelligkeit der Workflows und zieht Vergleiche zwischen den manuell erstellten und KI-gestützten Varianten. Kapitel fünf dient der Diskussion der Ergebnisse. Hier werden die Erkenntnisse kritisch reflektiert und in den Kontext der bisherigen Forschung gestellt. (Zudem wird die Forschungsfrage abschließend beantwortet und die Hypothese überprüft.) Abschließend bietet Kapitel sechs eine Zusammenfassung der zentralen Ergebnisse und einen Ausblick auf mögliche zukünftige Forschungsschwerpunkte.

1.4 Theoretischer Rahmen

Um die Leistungsfähigkeit zu analysieren, bietet das nächste Kapitel eine umfassende theoretische Grundlage. Es wird auf die Kerntechnologien und Konzepte von Microsoft Copilot für M365 und Microsoft Power Automate eingegangen. Dabei wird erklärt, wie diese Technologien funktionieren, auf welchen Komponenten sie aufbauen und wie sie zu handhaben sind. Es wird, abgesehen von den zwei Microsoft Produkten, auch auf die Technologie hinter den beiden Produkten eingegangen. Das wären bei Microsoft Copilot für M365, „Large Language Models (LLMs)“, bei Microsoft Power Automate ist die zugrundeliegende Technologie die „Robotic Process Automation (RPA)“.

Zuerst werden die technischen Grundlagen von Microsoft Copilot für M365 erläutert. Ein Augenmerk wird auf die Integration in die Microsoft 365-Umgebung und seine

Funktionalitäten gelegt. Dabei wird auf Large Language Models (LLMs), den Microsoft Graph und den Grounding-Prozess näher eingegangen.

Des Weiteren werden die Eigenschaften und die Funktionsweise von Microsoft Power Automate als zentrale Plattform für die Workflow-Automatisierung erläutert. Es wird beschrieben, wie Power Automate durch Low-Code-Ansätze eine benutzerfreundliche Erstellung und Verwaltung von Workflows ermöglicht und dabei sowohl Microsoft-365-Dienste als auch externe Systeme integriert. Die Verbindung zwischen Copilot und Power Automate schafft eine neue Ebene der Automatisierung, bei der KI eine unterstützende Rolle übernimmt.

Zusätzlich wird auf die Herausforderungen und Potenziale der KI-gestützten Workflow-Automatisierung eingegangen. Ein besonderer Fokus liegt hierbei auf der Effizienz und der Fehleranfälligkeit von KI-generierten Workflows. Es wird analysiert, inwiefern KI-gestützte Automatisierung dazu beitragen kann, repetitive Aufgaben zu optimieren, und welche Risiken durch mögliche Fehler oder Missverständnisse bei der Interpretation natürlicher Sprache bestehen.

Abgerundet wird der theoretische Rahmen durch eine Übersicht über bestehende Studien und Literatur, die den aktuellen Forschungsstand zur Effizienz und Fehleranfälligkeit von KI-gestützten Systemen dokumentieren. Diese Betrachtungen legen das Fundament für die experimentelle Fallstudie, die in den folgenden Kapiteln beschrieben wird.

2. Theoretische Grundlagen

In diesem Kapitel der Bachelorarbeit wird ein grundlegendes Verständnis über Microsoft Copilot, Microsoft Power Automate und die KI-gestützte Workflow-Automatisierung aufgebaut. Dieses Kapitel bildet das theoretische Fundament, auf dem die Fallstudie und die Analyse der Bachelorarbeit aufbauen.

Zuerst werden die Grundlagen von Microsoft Copilot und Microsoft Power Automate behandelt, um deren Funktionsweisen und Einsatzmöglichkeiten zu verstehen. Das darauffolgende Kapitel widmet sich der KI-gestützten Workflow Automatisierung, welche einen Überblick über das generelle Zusammenspiel von KI-Systemen und Workflow Automatisierungssoftwares gibt. Um auch einen theoretischen Einblick in die Bewertung zu bekommen, wird ein Unterkapitel auch die Themen Effizienz und Fehleranfälligkeit bei der KI-gestützten Programmierung behandeln. Abschließend werden relevante Studien und theoretische Modelle von bestehenden Arbeiten analysiert und interpretiert.

2.1 Grundlagen von Microsoft Copilot für M365

2.1.1 Einführung in den Microsoft Copilot für M365

Microsoft kann eine lange Geschichte in der Entwicklung intelligenter Assistenten aufweisen. Angefangen mit dem Office Assistent „Clippy“ bis hin zur heutigen Version von Microsoft Copilot für M365. Während Clippy noch stark eingeschränkte Funktionen hatte, ist Microsoft Copilot für M365 ein vollständig integrierter Assistent in der Microsoft 365-Umgebung (Stratton, 2024).

Dieser Assistent wurde in die Microsoft 365-Umgebung integriert, um die Produktivität zu steigern und Arbeitsprozesse zu erleichtern. Microsoft Copilot für M365 basiert auf fortschrittlichen Technologien, darunter Azure (Cloud Computing Plattform von Microsoft), OpenAI und dem Microsoft Graph. Diese Technologien helfen natürliche Sprache zu verarbeiten und kontextspezifische Inhalte zu generieren. Der Hauptfokus von Microsoft Copilot für M365 liegt darin, komplexe Aufgaben zu vereinfachen, indem er Benutzern ermöglicht, durch einfache Sprachbefehle Aktionen auszuführen. Sei es das Zusammenfassen von Emails oder Word Dokumenten, das Erstellen einer Präsentation in PowerPoint oder die Interpretation von größeren Datenmengen in Excel. Diese Fähigkeiten unterscheidet Copilot von traditionellen Automatisierungslösungen und anderen KI-Systemen (Stratton, 2024).

2.1.2 Technische Architektur und Funktionsweise

Stratton (2024) behauptet, dass Microsoft Copilot auf drei Schlüsseltechnologien basiert, welche die Nutzung von Microsoft Copilot gewährleisten. Wie diese drei Technologien

zusammenarbeiten, um von der Eingabe eines Prompts zur Ausgabe der Antwort oder zur Durchführung einer Aktion gelangen, wird in Abbildung 1 veranschaulicht.

2.1.2.1 Large Language Models

Microsoft Copilot für M365 basiert auf einem LLM, das auf Microsofts eigenen Servern unter den Azure OpenAI-Services gehostet wird. Hinter diesen OpenAI-Services befinden sich abgewandelte Versionen von OpenAI's Large Language Models. Die Einrichtung des LLM in Microsoft Azure sorgt dafür, dass das Copilot Modelle unabhängig von der allgemeinen OpenAI-Instanz funktionieren und speziell für die Microsoft 365-Umgebung angepasst sind. OpenAI hat somit keinen Zugriff auf die abgewandelten Sprachmodelle oder die Daten, die darin verarbeitet werden, was eine grundlegende Datensicherheit gewährleistet. Die Daten, die dem LLM zur Verfügung stehen, müssen trotzdem abgefragt werden, hier kommt der Microsoft Graph ins Spiel (Stratton, 2024).

2.1.2.2 Microsoft Graph

Der Microsoft Graph ist ein Application Programming Interface (API), welches Zugriff auf fast alle Daten in der Microsoft 365-Umgebung bietet, darunter E-Mails, Outlook Kalender, SharePoint Listen, Teams Chats, Dateien aus OneDrive oder auch Security Gruppen aus EntraID. Der Microsoft Graph bildet einen weiteren Grundstein für die Funktionalität von Copilot, da er alle erforderlichen Daten aus der M365-Umgebung abrufen kann. Dadurch können benutzerspezifische Informationen bereitgestellt werden, ohne sensible Daten zu speichern oder außerhalb der Microsoft-Umgebung zu verarbeiten. Diese Architektur erleichtert die Zusammenarbeit und den sicheren Datenzugriff. Hier unterliegt der Microsoft Graph dennoch den Berechtigungsstrukturen der Microsoft 365-Umgebung. Ein bestimmter Benutzer kann somit nur auf geteilte Dateien, für alle verfügbare Dateien und seine eigenen zugreifen, sofern diese in der Microsoft 365-Umgebung gespeichert sind (Stratton, 2024).

2.1.2.3 Graph-Grounded-Chat und Web-Grounded-Chat

Beim Grounding-Prozess gibt es zwei verschiedene Arten von Grounding, welche unterschiedliche Ziele verfolgen. Grounding ist grundsätzlich ein Prozess, bei dem das Sprachmodell mit spezifischem Wissen ausgestattet wird, welches einen Kontextbezug in einem bestimmten Rahmen bildet (Stratton, 2024).

Graph-Grounded Chats werden üblicherweise im Enterprise-Kontext verwendet, also in Unternehmen. Hier „grounded“ sich der Copilot über die Daten der gesamten Microsoft 365-Umgebung. Dabei wird sichergestellt, dass Microsoft Copilot für M365 ausschließlich auf vertrauenswürdige, unternehmensspezifische und benutzerspezifische Daten zugreift. Das Grounding schützt die Privatsphäre der Nutzer und ermöglicht Microsoft Copilot für M365, genaue und sichere Antworten zu generieren, die den aktuellen Datenschutzrichtlinien entsprechen. Außerdem kann Copilot im Graph-

Grounded-Chat Fragen im unternehmensspezifischen Kontext beantworten. Er ist in der gesamten Office-Suite integriert und kann für das Bewältigen spezifischer Aufgaben benutzt werden. Der Copilot kann auch direkt über den Chat in Teams konsultiert werden (Stratton, 2024).

Der Web-Grounded-Chat steht, wie man aus dem Namen ableiten kann, direkt im Microsoft Edge Browser oder unter <https://copilot.microsoft.com> zur Verfügung. Dieser Chat „grounded“ sich über das Internet, allen voran über die Bing Suche. Somit können auch Anfragen, die nicht unternehmensspezifischer Natur sind, beantwortet werden. Bis vor kurzem gab es die Möglichkeit Bilder zu erstellen nur in der Web-Grounded Version von Copilot. Mittlerweile ist dieses Feature auch im Copilot für M365 verfügbar (Stratton, 2024).

Zusammengefasst heißt das, dass Microsoft Copilot für M365 ein Graph-Grounded-Chat ist, welcher bei Fragen aus deiner Arbeitswelt unterstützen soll und Zugriff auf Unternehmensspezifische Daten hat. Wie schon im Namen ist dieser in die Microsoft Umgebung eingebettet. Microsoft Copilot dagegen ist ein Web-Grounded-Chat, welcher sich Ressourcen aus dem Internet bedient und hauptsächlich als persönlicher Assistent verwendet werden soll (Stratton, 2024).

In der Bachelorarbeit wird der Microsoft Copilot für M365 verwendet. Daher ist in den folgenden Kapiteln Microsoft Copilot für M365, Microsoft Copilot und nur Copilot, als dasselbe Produkt zu sehen.

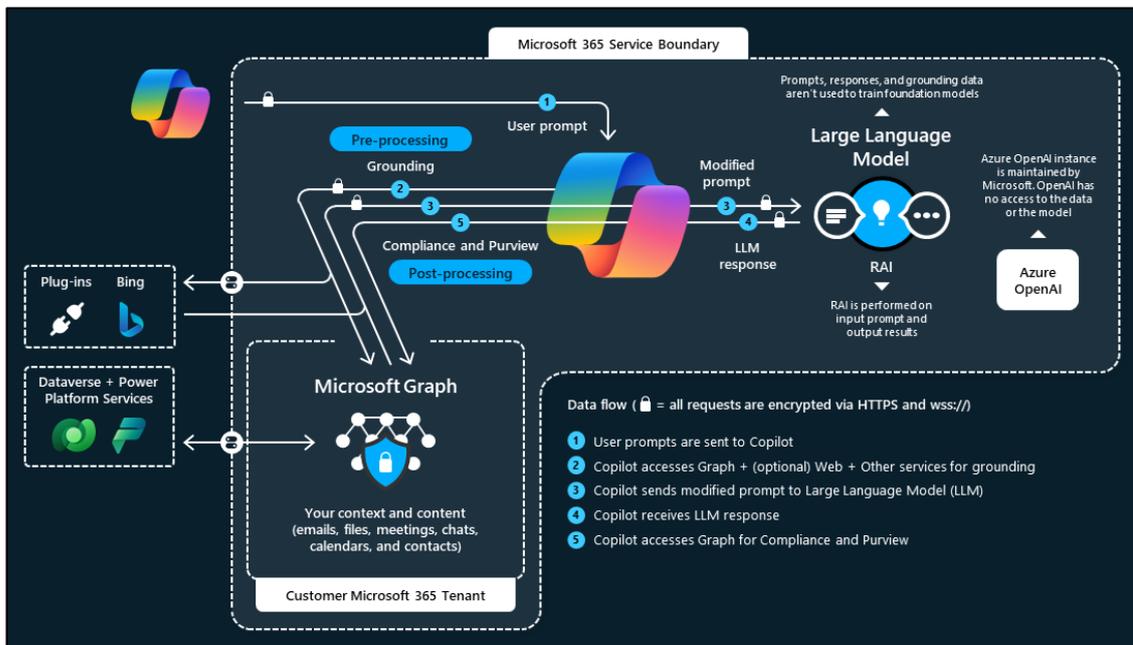


Abbildung 1: Copilot Prompt Verarbeitung (camillepack, 2024)

2.1.3 Grundlagen zu Large Language Models (LLMs)

Large Language Models sind eine zentrale Technologie im Bereich der künstlichen Intelligenz und bilden das Fundament vieler moderner KI-Anwendungen, einschließlich Microsoft Copilot. Sie ermöglichen es, große Mengen an Textdaten zu verarbeiten, zu analysieren und auf deren Grundlage menschenähnliche Antworten zu generieren. Dieses Kapitel vermittelt die grundlegenden Konzepte, den Aufbau und die Funktionsweise von LLMs, um ein besseres Verständnis für ihre Rolle in der Automatisierung und in KI-gestützten Systemen zu schaffen (Liu et al., 2025).

2.1.3.1 Grundlegender Aufbau von LLMs

LLMs basieren auf neuronalen Netzwerken, die speziell für die Verarbeitung von Sprache entwickelt wurden. Sie sind darauf trainiert, Muster und Zusammenhänge in Textdaten zu erkennen. Der Aufbau dieser Modelle beinhaltet mehrere Schichten, die jeweils eine spezifische Aufgabe erfüllen:

Eingabe (Input): LLMs arbeiten mit Textdaten, die in numerische Werte, sogenannte Vektoren, umgewandelt werden. Dies geschieht, damit der Text für das Modell verständlich wird.

Verarbeitung (Hidden Layers): Im Kern des Modells analysieren mehrere Verarbeitungsschichten die Bedeutung der Wörter im Kontext zueinander. Dies geschieht mithilfe eines Mechanismus namens „Transformer“ - dieser erkennt, wie Wörter in einem Satz miteinander zusammenhängen.

Ausgabe (Output): Nach der Verarbeitung generiert das Modell eine Antwort oder führt eine Aufgabe aus, basierend auf den Eingabedaten.

(Liu et al., 2025)

2.1.3.2 Wie funktionieren LLMs?

Die Funktionsweise von LLMs beruht auf zwei zentralen Prinzipien - Training und Generierung. LLMs werden mit großen Textdatensätzen trainiert, die aus Büchern, Artikeln und anderen Quellen stammen. Während des Trainings lernt das Modell, Wahrscheinlichkeiten für Wörter und Satzstrukturen zu berechnen. Zum Beispiel kann das Modell vorhersagen, welches Wort mit hoher Wahrscheinlichkeit als nächstes in einem Satz erscheinen sollte. Sobald ein LLM trainiert ist, kann es basierend auf einer Eingabe Inhalte generieren. Es analysiert die Eingabe, greift auf das Gelernte zurück und erstellt eine passende Antwort. Diese Fähigkeit macht LLMs flexibel und vielseitig einsetzbar, von der Textgenerierung bis zur Problemlösung (Liu et al., 2025).

2.1.3.3 Stärken von LLMs

Large Language Models zeichnen sich durch ihre Fähigkeit aus, große Mengen an Textdaten zu verarbeiten und kontextbezogene Antworten zu liefern. Sie können

Informationen zusammenfassen und prägnant darstellen, kreative Inhalte wie Geschichten oder Gedichte erstellen, komplexe technische Anfragen in einfache Sprache übersetzen. Diese Stärken machen LLMs zu einem unverzichtbaren Werkzeug in Anwendungen wie Microsoft Copilot, wo sie die Workflow-Automatisierung durch intuitive Benutzereingaben unterstützen (Liu et al., 2025).

2.1.3.4 Herausforderungen von LLMs

Trotz ihrer beeindruckenden Fähigkeiten gibt es auch Herausforderungen. Modelle können Fehlinformationen generieren, wenn die Eingaben ungenau oder mehrdeutig sind, Schwierigkeiten haben, seltene oder spezifische Kontexte korrekt zu verarbeiten oder benötigen hohe Rechenressourcen, was ihren Einsatz kostenintensiv machen kann (Liu et al., 2025).

2.1.3.5 Warum sind LLMs relevant für diese Arbeit?

Für Microsoft Copilot spielen LLMs eine Schlüsselrolle, da sie die Schnittstelle zwischen der Benutzeranfrage und der Workflow-Automatisierung bilden. Durch die Fähigkeit, natürliche Sprache zu verstehen und umzusetzen, können LLMs bei der Erstellung von Workflows unterstützen. Dieses Kapitel schafft die Grundlage, um die Effizienz und Fehleranfälligkeit solcher Systeme, wie in dieser Arbeit untersucht, besser zu verstehen (Liu et al., 2025).

2.1.4 Prompt Engineering

Prompt Engineering ist eine Technik, um die Leistung von LLMs wie ChatGPT oder Microsoft Copilot zu verbessern. Dabei werden Eingabebefehle (Prompts) so gestaltet und strukturiert, dass die generierten Antworten eher den Erwartungen der Anwender entsprechen. Diese Praxis gewinnt mit der zunehmenden Verbreitung von KI-gestützten Systemen an Bedeutung, da sie die Schnittstelle zwischen menschlicher Absicht und maschineller Ausgabe bildet (Ekin, 2023).

2.1.4.1 Die Rolle von Prompts in der Interaktion mit LLMs

Prompts sind die zentrale Kommunikationsschnittstelle zwischen Anwendern und LLMs. Sie beeinflussen die Qualität der generierten Inhalte. Ein klar formulierter Prompt führt zu präziseren Antworten. Eine unklare Eingabe auf der anderen Seite kann zu unerwünschten oder fehlerhaften Ergebnissen führen. Dabei spielen Aspekte wie die Spezifikation des Inhalts, die Länge und der Kontext eine entscheidende Rolle (Ekin, 2023).

2.1.4.2 Beispiel eines Prompts

Prompt 1 (Allgemein) : „Schreibe ein Gedicht“

Ausgabe 1: „In der Stille der Nacht, Ein einsamer Mond erhebt sich, Seine silbernen Strahlen streicheln das Land, Eine sanfte Berührung von einer himmlischen Hand.“

Prompt 2 (Spezifisch): „Schreibe ein Haiku über die sich ändernden Jahreszeiten“

Ausgabe 2: „Herbstblätter fallen langsam, Der Atem des Winters kühlt, Frühlingsknospen wachsen, Die Sommersonne strahlt.“

Man erkennt, dass die zweite Ausgabe spezifischer und eher dem entspricht, was der Anwender möchte, da im Prompt eine Art von Gedicht und ein Thema angegeben wurde (Ekin, 2023).

2.1.4.3 Techniken des Prompt Engineerings

Klarheit: Ein gut formulierter Prompt gibt explizite Anweisungen und vermeidet Mehrdeutigkeit.

Kontext: Die Einbindung von Beispielen oder spezifischen Anweisungen hilft dem LLM, die Erwartungen besser zu verstehen.

Verfeinerung: Die Optimierung eines Prompts durch wiederholtes Testen und Anpassen verbessert die Ergebnisse.

(Ekin, 2023)

2.1.5 Sicherheit und Datenschutzaspekte

Microsoft Copilot verarbeitet Daten ausschließlich innerhalb der Azure OpenAI-Infrastruktur, die strengen Sicherheitsstandards folgt. Alle Daten verbleiben innerhalb der Tenant-Grenzen der Microsoft 365-Umgebung. Dies bedeutet, dass sensible Informationen nicht außerhalb des Unternehmensnetzwerks verarbeitet werden, was ein hohes Maß an Datensicherheit gewährleistet. Die Architektur nutzt Verschlüsselung auf allen Datenebenen sowie strikte Zugriffskontrollen, um die Integrität der Daten sicherzustellen (Khan, 2024).

Microsoft Copilot für M365 ist so konfiguriert, dass er den Anforderungen internationaler Datenschutzbestimmungen wie der Datenschutz-Grundverordnung (DSGVO) entspricht. Das System führt regelmäßige Audits und Compliance-Checks durch, um sicherzustellen, dass alle gesetzlichen Vorgaben eingehalten werden. Diese Maßnahmen umfassen unter anderem die Minimierung der Datennutzung, die Anonymisierung personenbezogener Informationen und die Einhaltung von Löschrichtlinien (Khan, 2024).

2.2 Grundlagen Microsoft Power Automate

Dieses Kapitel behandelt die grundlegenden Konzepte und Funktionen von Microsoft Power Automate. Microsoft Power Automate ermöglicht es, viele Aufgaben innerhalb

von M365 zu automatisieren und die Effizienz ihrer Arbeitsabläufe zu steigern. Es wird auf die wichtigsten Komponenten und Bausteine von Power Automate eingegangen. Dazu gehört die Erstellung und Verwendung von Workflows und der Einsatz von Triggern und Aktionen. Darüber hinaus wird die Bedeutung von Bedingungen und logischen Verzweigungen erklärt, die die Anpassung an spezifische Anforderungen ermöglichen.

2.2.1 Einführung in Microsoft Power Automate

Microsoft Power Automate ist ein zentraler Bestandteil der Microsoft Power Plattform und ermöglicht die Automatisierung von Prozessen in einer Low-Code-Umgebung. In einem nächsten Kapitel wird auf die Bedeutung von Low-Code näher eingegangen. Power Automate ist wie die gesamte Power Plattform cloudbasiert. Power Automate unterstützt sowohl die Integration mit Microsoft-365-Diensten wie SharePoint, Teams und Outlook als auch die Anbindung an externe Systeme wie Salesforce, Docu-Sign oder ServiceNow. So können Anwender vielseitige Workflows erstellen, die sowohl einfache Benachrichtigungen als auch ausführliche, systemübergreifende Automatisierungen umfassen (Pearson et al., 2020).

Zusätzlich schreibt Guilmette (2020), dass Power Automate durch seine Integration in M365 besonders für Aufgaben wie Genehmigungsprozesse, Datenaktualisierungen oder Erinnerungen geeignet ist.

2.2.2 Funktionsweise von Microsoft Power Automate

Die Funktionsweise von Power Automate basiert auf Workflows, die aus **Triggern (Auslösern)** und **Aktionen** bestehen. Ein Trigger ist ein Ereignis, das den Workflow startet (auslöst). Das könnte zum Beispiel der Eingang einer neuen E-Mail in einem bestimmten Postfach, das Hinzufügen eines Eintrags in einer SharePoint-Liste, aber auch ein Klick auf eine Schaltfläche sein. Die darauffolgenden Aktionen sind Schritte, die nacheinander abgearbeitet werden. Aktionen reichen vom Versenden einer Nachricht bis zur Aktualisierung einer Variable durch eine Berechnungen. Ein Workflow beschreibt somit ein durch einen Trigger ausgelöste schrittweise Abarbeitung von Aktionen (Guilmette, 2020) (Pearson et al., 2020).

Microsoft Power Automate unterstützt verschiedene Typen von Workflows, darunter:

- **Automatisierte Workflows:** Starten automatisch durch ein bestimmtes Ereignis
- **Instant Workflows:** Werden immer manuell durch den Benutzer gestartet.
- **Geplante Workflows:** Werden zu einer festgelegten Zeiten gestartet
- **Desktop Workflows:** Ermöglichen die Automatisierung von Desktop-Anwendungen.

(Guilmette, 2020)

2.2.3 Low-Code

Power Automate ist eine Low-Code-Lösung, die es auch technisch weniger versierten Nutzern ermöglicht, Workflows zu erstellen. Die Plattform verwendet eine grafische Benutzeroberfläche, in der Aktionen und Auslöser per Drag-and-Drop angeordnet werden können. So wird die Abhängigkeit von Programmiererfahrung reduziert und eine schnellere Umsetzung von Automatisierungslösungen ermöglicht. Außerdem fördert der Low-Code-Ansatz die sogenannten "Citizen Developers". Dabei erstellen Anwender selbst Lösungen, ohne tiefgehende IT-Kenntnisse oder Unterstützung von IT-Personal zu benötigen (Guilmette, 2020) (Pearson et al., 2020).

2.2.4 Grundlagen zu Robot Process Automation (RPA)

Robotic Process Automation (RPA) oder robotergesteuerte Prozessautomatisierung ist eine Technologie für die Automatisierung von Abläufen, speziell für sich wiederholende und repetitive Aufgaben. Im Gegensatz zu traditionellen Automatisierungslösungen arbeitet robotergesteuerte Prozessautomatisierung auf der Benutzeroberfläche bestehender Anwendungen. Software-Roboter imitieren Aktivitäten von menschlichen Nutzern, indem sie auf Benutzeroberflächen zugreifen, Daten verarbeiten oder Aktionen ausführen und dadurch Aufgaben automatisieren. Der Vorteil hier ist, dass keine Änderungen an der zugrunde liegenden IT-Infrastruktur vorgenommen werden müssen (Hofmann et al., 2020).

RPA wird häufig in Back-Office-Prozessen eingesetzt, die eine hohe Wiederholungsrate aufweisen. Beispiele für solche Aufgaben sind das Ausfüllen von Formularen, das Kopieren und Einfügen von Datensätzen oder das Erstellen von Berichten. Software-Roboter können solche Prozesse selbst ausführen und dabei hohe Genauigkeit und Effizienz sicherstellen. Sibaliija et al. (2019) betonen, dass RPA besonders vorteilhaft für Aufgaben ist, die niedrige kognitive Anforderungen haben und einen hohen Automatisierungsgrad ermöglichen. RPA ist der zugrundeliegende Gedanke von Automatisierungslösungen wie Power Automate.

2.3 Grundlagen der KI-gestützten Workflow Automatisierung

Die Kombination von Künstlicher Intelligenz (KI) und Robotic Process Automation (RPA) ist ein entscheidender Schritt in der Entwicklung von Automatisierungslösungen. Während RPA normalerweise auf regelbasierten Prozessen basiert und repetitive Aufgaben automatisiert, erweitert die Integration von KI die Möglichkeiten erheblich. Durch die Verbindung beider Technologien entstehen intelligente Systeme, die in der Lage sind, sowohl strukturierte als auch unstrukturierte Daten zu verarbeiten, komplexe Entscheidungen zu treffen und Workflows adaptiv zu gestalten (Ribeiro et al., 2021).

KI-gestützte Workflow Automatisierung nutzt Technologien wie Natural Language Processing (NLP) und maschinelles Lernen. Somit können Aufgaben automatisiert werden, die über einfache, regelbasierte Prozesse hinausgehen. NLP ermöglicht es, natürliche Sprache zu analysieren und zu interpretieren, was insbesondere bei der Bearbeitung von Kundenanfragen oder der Analyse von E-Mail-Inhalten sehr hilfreich sein kann. Künstliche Intelligenzen können Muster in großen Datensätzen erkennen und kontinuierlich die Entscheidungsfindung der Automatisierung verbessern. Dadurch werden Workflows dynamischer und flexibler gestaltet. So kann die Effizienz erhöht, und die Anpassungsfähigkeit der Systeme gesteigert werden (Koch & Wildner, 2020).

Die Integration von Microsoft Copilot in Power Automate stellt eine solche Kombination von KI und Workflow Automatisierung dar. Microsoft Copilot verwendet KI-Technologien, um Benutzer bei der Erstellung von Workflows zu unterstützen, indem es natürliche Sprache versteht und automatisch Workflows generieren kann. Dies reduziert die Komplexität der Workflow-Erstellung und macht Automatisierung für eine breitere Menge an Nutzern zugänglich. Benutzer können einfache Anweisungen wie "Erstelle einen Workflow, der mich benachrichtigt, wenn eine neue Datei hinzugefügt wird" eingeben, und Copilot in Kombination mit Power Automate erstellt den entsprechenden Workflow dann automatisch (v-aangie, 2024).

2.4 Effizienz und Fehleranfälligkeit bei der KI-gestützten Programmierung

Basierend auf Erkenntnissen zur KI-gestützten Programmierung wie zum Beispiel GitHub Copilot lassen sich Parallelen zur Workflow-Automatisierung ziehen. Insbesondere steht die Art der Interaktion zwischen Anwendern und KI-Systemen im Fokus.

2.4.1 Effizienz durch KI-gestützte Automatisierung

KI-Systeme wie GitHub Copilot basieren auf großen Sprachmodellen (LLMs) und bieten Vorteile in der Effizienzsteigerung, indem sie repetitive Aufgaben automatisieren und Vorschläge zur Lösung spezifischer Probleme machen. Die Studie von Barke et al. (2022) hebt hervor, dass die Interaktion zwischen Anwender und KI-Systemen auf zwei verschiedene Weisen stattfindet: "Acceleration (Beschleunigung)" und "Exploration (Erforschung)". Im Beschleunigungs-Modus können Nutzer klar definierte Aufgaben schneller lösen, indem die KI Vorschläge für spezifische Arbeitsschritte liefert oder den Code für eine bestimmte Funktion. Diese Funktionalität entspricht der Automatisierung einfacher Workflow-Schritte, wie dem Trigger eines Workflows oder dem Senden einer Mail als Einzelfunktion. Im Erforschungs-Modus wissen Nutzer noch nicht genau, wie sie ein Problem lösen können, und lassen sich Vorschläge von der KI generieren, um dann einen Ansatz zu wählen und diesen weiterzuverfolgen. Diese Funktionalität kann dabei

helfen, sich über das weitere Vorgehen klar zu werden. In der Workflow-Automatisierung kann das helfen, wenn man noch keine Vorstellung davon hat, wie ein potenzieller Workflow aussehen kann (Barke et al., 2022).

2.4.2 Herausforderungen durch Fehleranfälligkeit

Trotz der Effizienzvorteile bleiben Herausforderungen bestehen, insbesondere bei der Validierung der von der KI generierten Inhalte. Die Fehleranfälligkeit steigt in erforschenden Szenarien, in denen Nutzer sich nicht sicher sind, wie sie weiter vorgehen sollen. Programmier-Anwender neigen dazu, generierte Vorschläge intensiver zu prüfen, wenn sie mit unbekanntem Aufgaben oder Schnittstellen arbeiten sollen (Barke et al., 2022).

2.4.3 Daten zur Effizienz und Validierung

Die Studie von Barke et al. (2022) liefert wertvolle empirische Daten zur Effizienz und Fehleranfälligkeit bei der Interaktion von Nutzern mit KI-gestützten Automatisierungssystemen wie GitHub Copilot. Die Ergebnisse zeigen deutliche Unterschiede in der Zeitverteilung und den Validierungsstrategien zwischen dem Beschleunigungs-Modus und dem Erforschungs-Modus.

Die Teilnehmer verbrachten insgesamt 248,6 Minuten im explorativen Modus und 104,7 Minuten im Beschleunigungsmodus. Dabei stellt sich heraus, dass professionelle Programmierer mit Erfahrung in GitHub Copilot häufiger im Beschleunigungs-Modus interagierten, was auf ihre Fähigkeit hinweist, klar definierte Aufgaben effizient umzusetzen. Unerfahrene Nutzer auf der anderen Seite verbrachten mehr Zeit im Erforschungs-Modus, was der Notwendigkeit einer intensiveren Validierung zu Grunde liegt (Barke et al., 2022).

Zusätzlich wurden drei Hauptmethoden zur Validierung in der Studie identifiziert:

Sorgfältige Prüfung des Codes: 57,2 % der Validierungszeit wurden für die genaue Analyse der Vorschläge verwendet.

Ausführung von Code: Nutzer führten die generierten Inhalte aus, um deren Funktionalität zu überprüfen.

Vergleich mit Dokumentationen: KI-Vorschläge wurden mit offiziellen Quellen abgeglichen, um die Korrektheit sicherzustellen.

(Barke et al., 2022)

2.5 Literaturanalyse

Die Literatur zu KI-gestützter Workflow-Automatisierung zeigt einige Perspektiven und Ansätze, die zur Bewertung der Effizienz und Fehleranfälligkeit dieser Systeme dienen. Dieses Kapitel beschäftigt sich mit Erkenntnissen aus drei wissenschaftlichen Papers. Diese decken sowohl theoretische als auch praktische Aspekte der Automatisierung und der Rolle von KI ab. Die ausgewählten Arbeiten ergänzen die Themen und Ziele dieser Bachelorarbeit, indem sie relevante Konzepte wie die Integration von KI in Robotic Process Automation (RPA), die Funktionalität von Large Language Models (LLMs) und die Herausforderungen bei der Nutzung von KI-Systemen erläutern.

2.5.1 Die Rolle von LLMs in der Automatisierung

Das erste Paper von Pandey et al. (2024), analysiert, wie Large Language Models (LLMs) die Automatisierung in Unternehmen revolutionieren können. Insbesondere wird darauf eingegangen, wie LLMs durch ihre Fähigkeit, natürliche Sprache zu verstehen und zu verarbeiten, eine benutzerfreundliche Schnittstelle zur Erstellung von Workflows bieten. Dies ist relevant für die Nutzung von Microsoft Copilot in Power Automate, da es Benutzern ermöglicht, Workflows durch einfache sprachliche Eingaben zu generieren. Die Studie hebt hervor, dass die Hauptstärken von LLMs in ihrer Flexibilität und ihrem adaptiven Lernen liegt, wodurch sie sich an unterschiedliche Kontexte anpassen können. Sie warnt jedoch auch vor Herausforderungen wie der Interpretation von Mehrdeutigkeiten in Benutzereingaben, was potenziell zu Fehlern in der Workflow-Erstellung führen kann (Pandey et al., 2024).

2.5.2 Innovationen in der RPA durch KI

Die Arbeit von Afrin et al. (2025), untersucht, wie KI-Technologien, einschließlich LLMs, RPA weiterentwickeln. Sie beschreibt, wie traditionelle RPA-Systeme durch die Integration von KI flexibler und leistungsfähiger werden. Diese Erkenntnisse stehen in direktem Zusammenhang mit Microsoft Power Automate, das als Low-Code-Plattform RPA-Funktionalitäten bietet. Laut der Studie ermöglichen KI-gestützte Systeme eine dynamischere Anpassung von Workflows und eine effizientere Verarbeitung großer Datenmengen. Gleichzeitig wird betont, dass die Einführung solcher Technologien nicht ohne Risiken ist. Zu den Hauptproblemen zählen die Abhängigkeit von qualitativ hochwertigen Trainingsdaten und die potenzielle Fehleranfälligkeit bei unvorhergesehenen Eingaben (Afrin et al., 2025).

2.5.3 Effizienz und Fehleranfälligkeit von KI-gestützten Workflows

Simao & Shaddoc (2023) konzentrieren sich auf die Bewertung der Effizienz und Fehleranfälligkeit von KI-gestützten Automatisierungssystemen. Sie bietet wertvolle

Erkenntnisse zur Messung und Bewertung dieser beiden zentralen Kriterien, die auch in dieser Bachelorarbeit untersucht werden. Die Studie stellt fest, dass KI-Systeme zwar eine schnellere Implementierung und Ausführung von Workflows ermöglichen, aber auch eine höhere Fehleranfälligkeit aufweisen können, insbesondere, wenn komplexe oder schlecht definierte Aufgaben automatisiert werden sollen. Die Ergebnisse der Studie bestätigen die Hypothese dieser Arbeit, dass KI-gestützte Workflows zwar effizienter sind, aber eine höhere Fehleranfälligkeit im Vergleich zu manuellen Workflows aufweisen (Simao & Shaddoc, 2023).

2.5.4 Relevanz für die Bachelorarbeit

Die analysierten Studien bieten eine solide Grundlage, um die Forschungsthemen dieser Arbeit zu unterstützen. Sie zeigen, dass LLMs und KI-gestützte RPA-Systeme sowohl Potenziale als auch Herausforderungen in der Workflow-Automatisierung mit sich bringen. Insbesondere die Erkenntnisse zur Effizienz und Fehleranfälligkeit sowie zur Benutzerfreundlichkeit von KI-basierten Systemen tragen dazu bei, die in dieser Arbeit verwendeten Bewertungskriterien zu validieren und ein besseres Verständnis für die Leistungsfähigkeit von Microsoft Copilot in der Praxis zu entwickeln.

3. Methodik

Dieses Kapitel stellt den Übergang von Theorie zum empirischen Arbeiten dar. Zuerst wird auf die empirische Methode eingegangen, welche für die Bachelorarbeit verwendet werden. Danach werden die Use-Cases ausführlich beschrieben. Anschließend wird auf die Bewertungskriterien eingegangen und beschreiben, wie Messungen im Rahmen der experimentellen Fallstudie vorgenommen werden. Für den empirischen Teil werden alle Use-Cases in Power Automate jeweils mit Hilfe von Copilot und manuell erstellt und ausgeführt. Dabei wird die Berechtigungen eines Globalen Admins verwendet. Auf mögliche Berechtigungsunterschiede bei den einzelnen Workflows wird in der Diskussion eingegangen.

3.1 Auswahl der Use-Cases

In der experimentellen Fallstudie werden fünf Workflows jeweils manuell und mithilfe der Unterstützung von Microsoft Copilot erstellt. Dabei werden Messungen getätigt, die am Ende Aufschluss über die Leistungsfähigkeit von Microsoft Copilot und Microsoft Power Automate geben sollen. Dabei werden angefragte Use-Cases aus dem Arbeitskontext herangezogen.

Use-Case 1:

Eine Person 1 ist zuständig für eine Intranet-Seite in einer SharePoint-Umgebung. Diese Person 1 arbeitet mit mehreren anderen Personen an dieser Seite. Person 1 möchte immer per Mail informiert werden, wenn jemand ein neues Dokument in einem bestimmten Ordner ablegt. Die Mail soll den Namen der Person, die die Datei hochgeladen hat und den Namen der Datei enthalten.

Use-Case 2:

Wenn eine Mail an eine bestimmte Inbox zugestellt wurde, soll durch Copilot der Inhalt der Mail bewertet und kategorisiert werden. Nach der Bewertung sollen mögliche Aufgaben, die identifiziert wurden, in Microsoft Tasks hinzugefügt werden. Dabei sind der Titel der Aufgabe, die auszuführende Tätigkeit falls näher beschrieben, die gewünschte Fertigstellung und zusätzliche Anhänge oder Ressourcen einzutragen.

Use-Case 3:

Eine Person übersieht öfters Neuigkeiten (News) aus ihrer Geschäftseinheit die am SharePoint gepostet wurden. Diese Neuigkeiten sind meistens lange Texte, die etwas Zeit in Anspruch nehmen, um sie zu lesen. Da die Person dennoch auf dem neusten

Stand bleiben möchte, möchte sie eine kurze zusammengefasste Version der Neuigkeiten per Mail erhalten, wenn eine neue News auf der Seite der Geschäftseinheit veröffentlicht wird. Der Workflow soll ihr als jedes Mal, wenn eine neue News veröffentlicht wird die zusammengefasste Version per Mail zukommen lassen.

Use-Case 4:

Eine Person betreibt eine SharePoint Liste für den Zugriff zu einem Teams-Team. Wenn man Zugriff zu diesem Teams-Team haben will, muss man sich mit seiner Email Adresse in diese Liste eintragen. Wenn sich eine Person einträgt, will die Person, die das Teams-Team betreibt, eine Benachrichtigung mit einem Approval erhalten. Wenn sie einverstanden ist, dass die Person Zugriff zu dem Teams-Team bekommt, bestätigt sie den Approval. Danach soll die Person automatisch zum Teams-Team hinzugefügt werden.

Use-Case 5:

Ein System-Administrator möchte, wenn neue Personen in der IT eingestellt werden, diese Personen in ein Teams-Team hinzufügen, um sich die manuelle Arbeit zu ersparen. Alle User der IT werden einer bestimmten Gruppe in EntraID zugeordnet. Der Systemadministrator möchte jedes Mal, wenn eine neue Person zu dieser EntraID Gruppe hinzugefügt wird, ihr auch den Zugriff auf das Teams-Team geben. Da Teams-Teams keine Gruppen als User unterstützen, würde ihm das einen Mehrwert bieten.

3.2 Experimentelle Fallstudie

In dieser Arbeit wird eine Kombination aus einer Fallstudie und einem Experiment angewandt, um die Forschungsfrage zu beantworten. Der gewählte Ansatz ermöglicht es, sowohl die reale Anwendung von Microsoft Copilot in einem praktischen Kontext zu untersuchen als auch Zusammenhänge unter kontrollierten Bedingungen zu evaluieren.

3.2.1 Die Fallstudie

Eine reine Fallstudie wäre primär deskriptiv und würde sich auf die Analyse eines spezifischen Anwendungsfalls in seinem natürlichen Kontext konzentrieren. Während dies nützliche Einblicke in die tatsächliche Nutzung von Microsoft Copilot und Power Automate liefern würde, wäre es schwierig, klare Ursache-Wirkungs-Zusammenhänge zwischen den eingesetzten Methoden und den Ergebnissen zu identifizieren.

3.2.2 Das Experiment

Ein reines Experiment erfordert die vollständige Kontrolle über alle Rahmenbedingungen, um präzise Kausalbeziehungen nachzuweisen. Dies ist jedoch in der Praxis oft schwer umzusetzen, da die realen Bedingungen, unter denen Workflows erstellt und verwendet werden, stark variieren. Experimente allein könnten daher die Komplexität und Vielschichtigkeit der tatsächlichen Anwendung von Microsoft Copilot nicht richtig abbilden.

3.2.3 Die Experimentelle Fallstudie

Der Mix aus Fallstudie und Experiment vereint die Stärken beider Methoden. Sie liefert Realitätsnähe durch Einblicke in die praktische Nutzung und ermöglicht eine Bewertung im realen Anwendungskontext. Zusätzlich erlaubt das experimentelle Design es, spezifische Variablen gezielt zu manipulieren und deren Auswirkungen auf die Effizienz und Fehleranfälligkeit von Workflows zu messen.

Dieser kombinierte Ansatz stellt sicher, dass sowohl praxisrelevante Erkenntnisse als auch wissenschaftlich fundierte Aussagen über die Leistungsfähigkeit von Microsoft Copilot im Vergleich zu manuell erstellten Workflows gewonnen werden können.

3.3 Bewertungskriterien

Die Bewertungskriterien beschreiben die zu messenden Größen und wie diese zu interpretieren sind. Dabei wird auch darauf eingegangen, wie die Dokumentation der Messgrößen erfolgt. Auf Basis dieser Bewertungskriterien werden die manuell erstellten Workflows mit den Copilot generierten Verglichen und in der Diskussion näher behandelt.

3.3.1 Fehleranfälligkeit

Die Fehleranfälligkeit beschreibt die Anzahl an fehlerhaft bzw. gescheiterte Workflow-Ausführungen, die sich ereignen, bis der Workflow die Aufgabenstellung vollständig erfüllen kann. Die Fehleranfälligkeit wird manuell erhoben und in der Excel-Datei dokumentiert. Um zu gewährleisten, dass die gezählte Zahl korrekt ist, gibt es noch eine Ausführ-Historie, in der die erfolgreichen sowie erfolglosen Ausführungen des Workflows mitgeschrieben werden, mit welcher die manuell erhobene Zahl verglichen wird. Fehler unterscheiden sich in 3 Kategorien, schwere Fehler, mittelschwere Fehler und leichte Fehler.

Schwere Fehler sind Fehler, bei denen der Workflow ohne manuelles eingreifen kein sinnvolles Ergebnis liefert. Zum Beispiel wenn ein Feld in einer Aktion nicht vom Copiloten ausgefüllt werden kann. Diese Fehler werden mit -3 Fehlerpunkten bewertet.

Ein **mittelschwerer Fehler** liegt vor, wenn der Workflow zwar grundsätzlich funktioniert, jedoch Teilergebnisse fehlerhaft oder unvollständig sind. Ein Beispiel wäre ein falsches Zuordnen von Daten oder das Auslassen optionaler, aber relevanter Schritte. Diese Fehler werden mit -2 Fehlerpunkten bewertet.

Leichte Fehler sind kleinere Abweichungen, die die Grundfunktionalität des Workflows nicht beeinträchtigen. Dies kann z. B. ein fehlender Kommentar oder eine geringfügige Formatierungsabweichung sein. Diese Fehler werden mit -1 Fehlerpunkten bewertet.

Ein Workflow gilt als weniger fehleranfällig, wenn er wenige oder keine fehlerhaften Ausführungen aufweist, bevor die Aufgabenstellung vollständig erfüllt ist. Dieses Kriterium ermöglicht es, die Qualität und Robustheit der Workflows zu beurteilen.

3.3.2 Schnelligkeit (Durchlaufzeit)

Die Schnelligkeit beschreibt die Durchlaufzeit des Workflows in Sekunden, gemessen ab dem Start des Triggers bis zur vollständigen Erfüllung der Aufgabenstellung. Diese Durchlaufzeit veranschaulicht die Schnelligkeit der Workflow-Ausführung.

Die Durchlaufzeit wird, ebenso wie die Ausführ-Historie von Microsoft Power Automate, bei jeder Ausführung dokumentiert und danach ebenso in die Excel-Tabelle übertragen, um alle erhobenen Daten vergleichen zu können.

Die Schnelligkeit eines Workflows bezieht sich auf die Zeitspanne zwischen dem Start des Triggers und der vollständigen Erfüllung der Aufgabenstellung. Eine kürzere Durchlaufzeit wird als positiver Indikator für die Schnelligkeit der Ausführung gewertet, eine lange Ausführung als ein negativer Indikator. Dieses Kriterium hilft, die Leistungsfähigkeit der Workflows im Betrieb zu vergleichen.

3.3.3 Effizienz

Die Effizienz beschreibt die Anzahl der Minuten, die benötigt werden, um einen Workflow vollständig und funktionstüchtig zu erstellen. Dabei wird untersucht, wie effizient die Ressourcen (Zeit) für die Erstellung des Workflows genutzt werden können .

Diese Dauer in Minuten wird manuell erfasst und in einem Excel Dokument nachvollziehbar dokumentiert. Für jeden Workflow wird jeweils das Datum der Erstellung sowie die Startzeit und Endzeit dokumentiert. Falls ein Workflow an mehreren Tagen erstellt wird, werden mehrere Tage mit Start- und Endzeit dokumentiert.

Dabei ist zu beachten, dass die Ergebnisse auf der Leistung einer einzelnen Person basieren - in diesem Fall des Autors dieser Arbeit. Die Entscheidung, nur die eigene Leistung als Maßstab zu verwenden, basiert auf der Absicht, die Zeitersparnis durch die Unterstützung von Microsoft Copilot als aussagekräftige Metrik zu betrachten, ohne dabei universelle oder allgemeingültige Aussagen zu treffen. Die Effizienz eines

Workflows wird daran gemessen, wie schnell er erstellt werden kann. Ein effizienter Workflow zeichnet sich durch einen kürzeren Erstellprozess aus, wobei das Ziel darin besteht, Zeit und Aufwand zu minimieren. Dieses Bewertungskriterium liefert Aufschluss darüber, ob und wie effizient Microsoft Copilot im Vergleich zur manuellen Erstellung Zeit spart.

4. Durchführung der Fallstudie

Zur praktischen Untersuchung der Leistungsfähigkeit von Microsoft Copilot in Power Automate wurde im Rahmen dieser experimentellen Fallstudie fünf praxisnahen Use-Cases herangezogen. Jeder dieser Use-Cases wurde als Workflows sowohl manuell als auch KI-gestützt mit Microsoft Copilot erstellt. Zur besseren Veranschaulichung des Vorgehens wurde vorab ein beispielhafter Workflow-Prozess dargestellt, der den Ablauf bei der Erstellung mit Copilot zeigt.

Für beide Varianten, manuell und KI-gestützt, wurden die durchschnittliche Durchlaufzeit der Workflows nach zehn erfolgreichen Ausführungen (bereinigt um Ausreißer mittels Standardabweichung), sowie die Fehlerpunkte gemäß dem in Kapitel 3.3.1 beschriebenen Bewertungsschema und die Dauer der Erstellung dokumentiert. Die folgenden Abschnitte stellen die Ergebnisse pro Use-Case dar und vergleichen diese miteinander.

4.1 Praktisches Beispiel: Erstellung eines Workflows mit Microsoft Copilot

Zur Veranschaulichung des Vorgehens bei der Workflow-Erstellung mit Microsoft Copilot wird in diesem Abschnitt ein konkretes Ablaufbeispiel beschrieben. Dabei wird Use-Case 1 exemplarisch anhand zweier Prompt-Varianten gegenübergestellt: der ursprünglichen Use-Case-Definition sowie einem angepassten Prompt, der zur erfolgreichen Erstellung des Workflows führte.

4.1.1 Ausgangslage und Ziel

In diesem Beispiel soll ein Workflow erstellt werden, der automatisch eine E-Mail an eine bestimmte Person sendet, sobald ein neues Dokument in einem spezifischen Ordner auf einer SharePoint-Seite hochgeladen wird. Die E-Mail soll den Namen der hochgeladenen Datei sowie den Namen der Person enthalten, die diese Datei hochgeladen hat.

4.1.2 Ausgangs Prompt (Use-Case Definition)

Zur Umsetzung mit Microsoft Copilot wurde der ursprüngliche Prompt aus der Use-Case-Definition in das Copilot-Feld eingegeben, wie in Abbildung 2 ersichtlich ist.

Prompt: „Eine Person 1 ist zuständig für eine Intranet-Seite in einer SharePoint-Umgebung. Diese Person 1 arbeitet mit mehreren anderen Personen an dieser Seite. Person 1 möchte immer per Mail informiert werden, wenn jemand ein neues Dokument in einem bestimmten Ordner ablegt. Die Mail soll den Namen der Person, die die Datei hochgeladen hat und den Namen der Datei enthalten.“

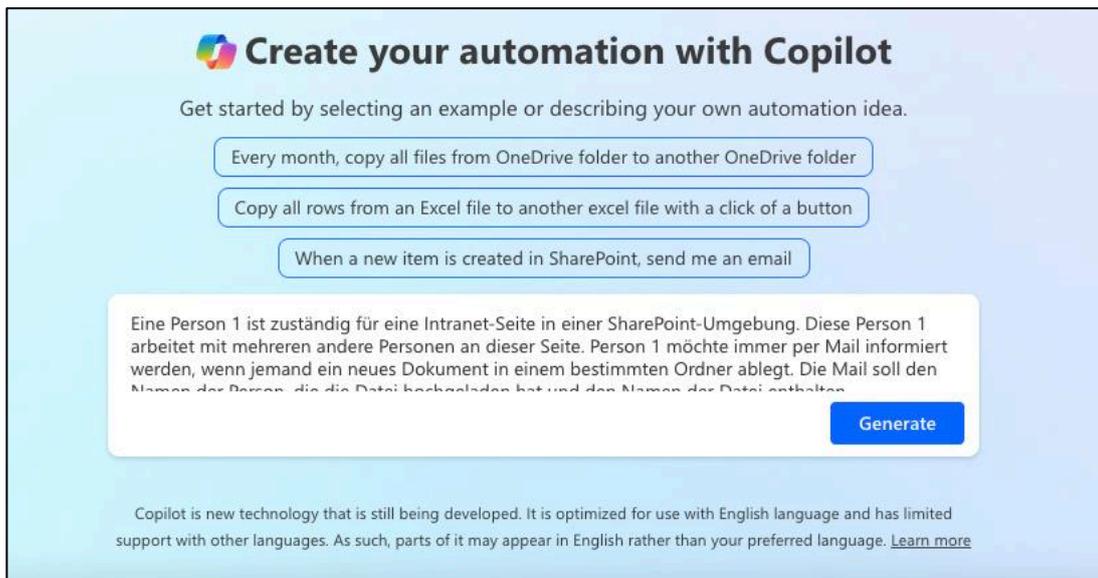


Abbildung 2: Copilot Erstellung mit Use-Case Definition

Mit einem Klicken auf den „Generate“ Knopf, wird, wie in Abbildung 3 zu sehen ist, ein möglicher Workflow vorgeschlagen. In diesem Fall wurden zwei potenzielle Versionen vorgeschlagen. Es wurde die erste Option gewählt, da die zweite schon beim Trigger einen Fehler aufwies und daher unbrauchbar war. Wenn man auf „Keep it and continue“ klickt, wird man mit dem vorgeschlagenen Workflow direkt in den Workflow-Builder von Power Automate weitergeleitet. Von dort aus können zusätzliche Einstellungen vorgenommen werden.

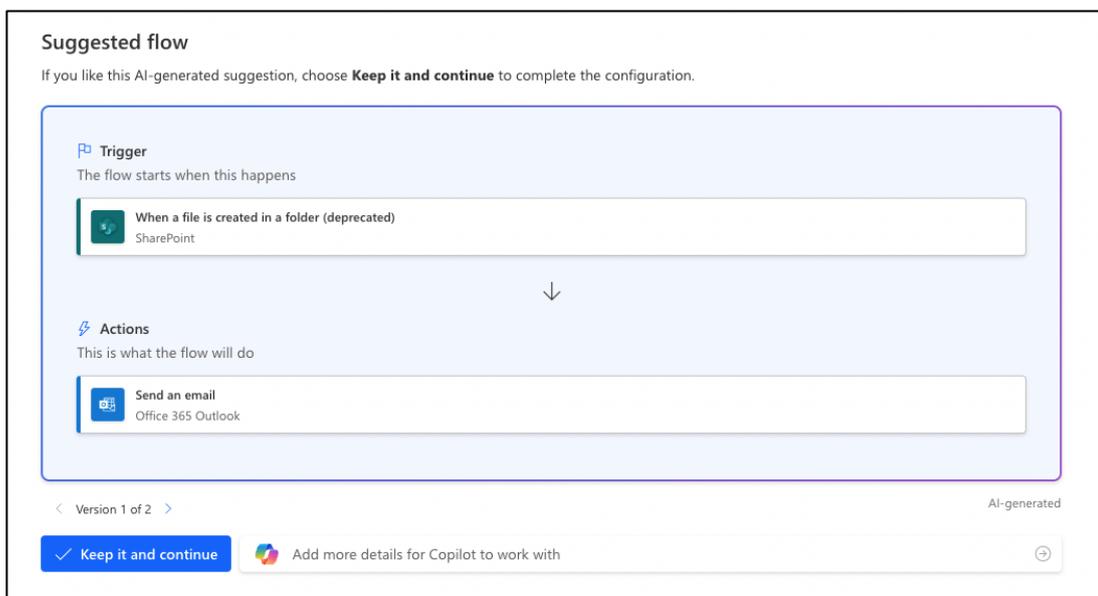


Abbildung 3: Copilot generierter Workflow mit Use-Case Definition

Wie man in Abbildung 4 sehen kann, war das Ergebnis auf Basis dieses Prompts nur teilweise korrekt. Der Trigger wurde erkannt und die E-Mail-Komponente grundsätzlich

erstellt. Jedoch fehlten die konkreten dynamischen Inhalte, wie Dateiname und Name der Person, die die Datei hochgeladen hat sowie spezifische Verweise auf die SharePoint-Seite und die Dokumenten-Bibliothek.

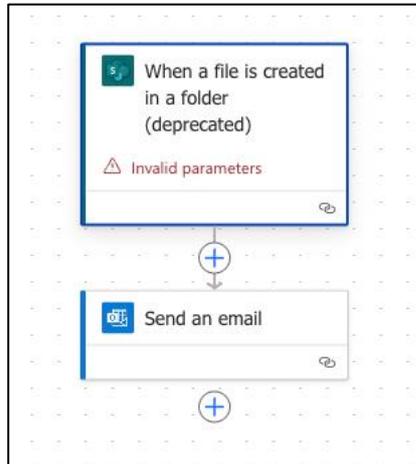


Abbildung 4: Unkonfigurierter Workflow

4.1.3 Finaler Prompt

Um ein genaueres Ergebnis zu erzielen wurde der Prompt überarbeitet und dann erneut wie in Abbildung 5 zu sehen ist in das Copilot Eingabefenster eingegeben und ausgeführt.

Prompt nach Überarbeitung: „Wenn eine Datei in den ‚Dokumenten‘ von dieser SharePoint Site `https://contoso.sharepoint.com` erstellt wird, dann sende eine Mail v2. Sende die Mail an „johannes.lienbacher@contoso.com“ (unter Verwendung der Person aus dem Dropdown). Der Betreff lautet „Neue Datei erstellt oder geändert“. Der Parameter „emailMessage/Body“ der Mail sollte eine Nachricht enthalten, dass eine neue Datei hochgeladen wurde, einschließlich des Erstellers der Datei „Modified By“ oder „Created By“ und der Dateiname der hochgeladenen Datei! Füge diese Informationen über dynamischem Contents hinzu! “

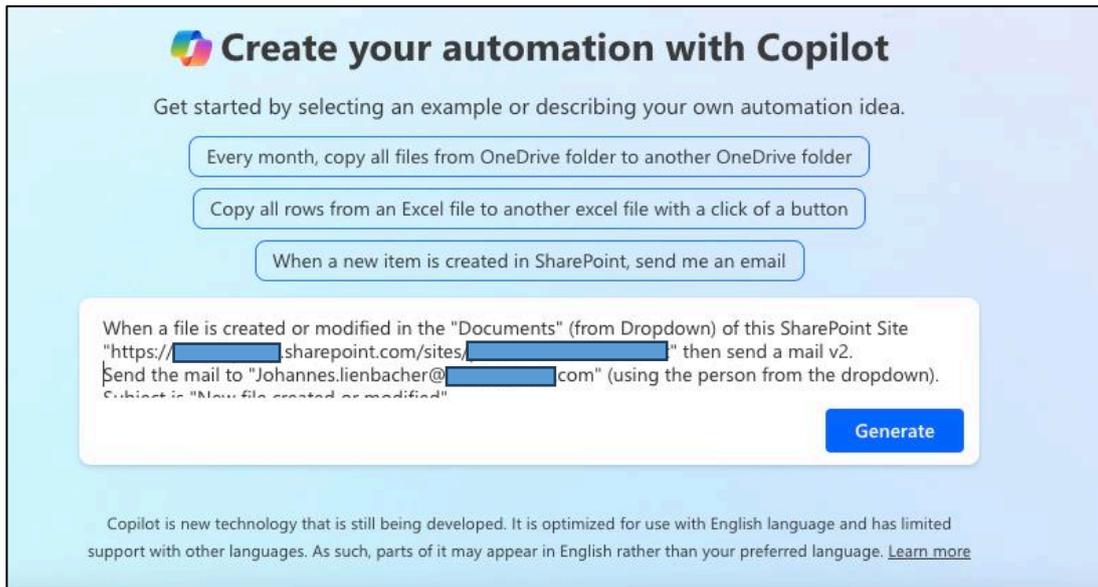


Abbildung 5: Copilot Erstellung mit eigenem Prompt

In Abbildung 6 ist der vorgeschlagenen Workflow von Copilot zu sehen. Auf dem ersten Blick sieht er exakt gleich aus wie zuvor. Die Änderungen des Prompts werden erst im Flow-Builder ersichtlich, da sich der Ablauf des Workflows nicht ändert, jedoch zusätzliche Konfiguration eingebaut wurde.

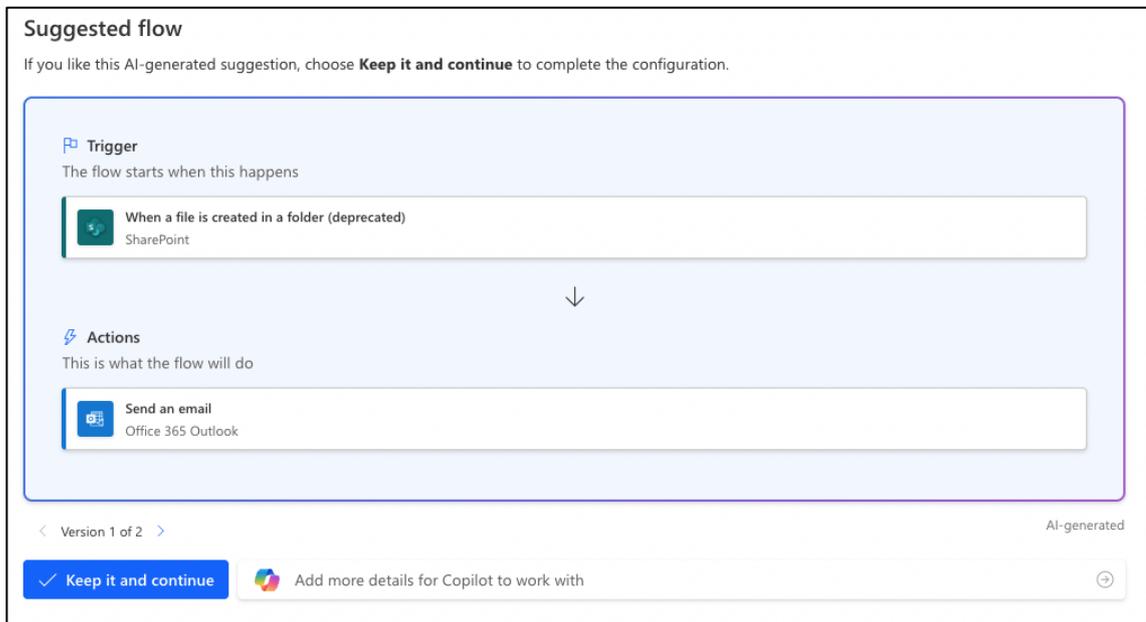


Abbildung 6: Copilot generierter Workflow mit eigenem Prompt

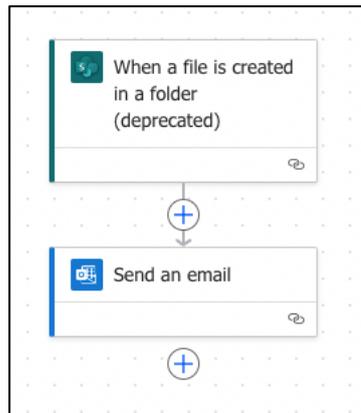


Abbildung 7: Teils Konfigurierter Workflow

Abbildung 7 zeigt, dass dieses Mal nach Erstellung des Flows keine Fehlermeldung erscheint. Um den Unterschied noch einmal genau darzustellen, wurde in den folgenden zwei Abbildungen (8 & 9) die Konfiguration der beiden Trigger-Aktionen gegenübergestellt.

Abbildung 8: Aktions-Konfiguration mit Use-Case Definition

Abbildung 9: Aktions-Konfiguration mit eigenem Prompt

Wie man in Abbildung 8 und Abbildung 9 erkennen kann, unterscheiden sich die Prompts in der Konfiguration der Aktionen. Umso genauer man die Konfiguration beschreibt,

desto präziser werden die von Copilot in Power Automate erstellten Workflows und umso weniger muss manuell oder mit dem „In-Flow-Builder-Chat“ ergänzt bzw. geändert werden. Dennoch zeigen sich auch bei detaillierten Prompts Grenzen von Copilot, die in den folgenden Kapiteln näher erläutert werden.

4.1.4 Anpassung mit Copilot im Flow-Builder

Die Trigger Aktion ist somit vollständig konfiguriert und einsatzbereit. Wenn nun der Workflow gespeichert wird, dann gibt es dennoch einen Error. Dies liegt daran, dass in der zweiten Aktion „Send an email“, der E-Mail-Body nicht korrekt befüllt wurde. Um so etwas anzupassen, kann die Copilot-Chat-Funktion direkt im Flow-Builder verwendet werden. In diesem Chat können Änderungswünsche beschrieben und dann von Copilot ausgeführt werden. Um den Workflow erfolgreich fertigzustellen, wurde der untenstehende Prompt verwendet. Abbildung 10 zeigt das Eingabefenster von Copilot im Flow-Builder.

Prompt: „Bitte ändere den Text des E-Mail-Bodys zu einem dynamischen Inhalt aus Aktion 1. Dabei sollen der Name der Datei und der Ersteller der Datei im Body angegeben werden.“

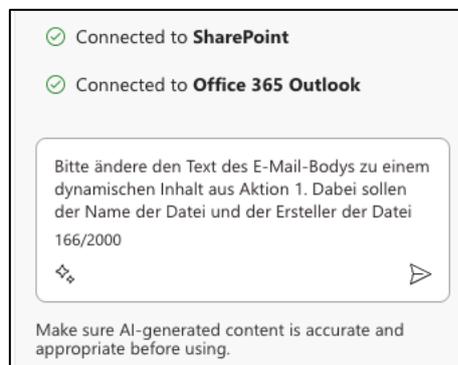


Abbildung 10: Copilot im Flow-Builder

In manchen Fällen kann Copilot bestimmte Prompts nicht korrekt verarbeiten. In diesem konkreten Beispiel brauchte es einige Versuche mit dem Flow-Builder-Chat, um die Funktion erfolgreich zu konfigurieren. Oftmals wurden ungewollte Aktionen gelöscht, unnötige Schleifen eingefügt, oder die gewünschte Konfiguration gänzlich ignoriert. Das Problem hierbei sind die dynamischen Inhalte, die es Power Automate ermöglichen, Metadaten aus einer vorherigen Aktion wiederzuverwenden. Dynamische Inhalte lassen sich in ihrer Funktion mit klassischen Variablen oder Arrays aus der Programmierung vergleichen. Dabei stehen sie für Werte, die sich je nach Kontext automatisch ändern. Konkret handelt es sich um den Dokumenten- und Personennamen, den das Dokument hochgeladen hat. Bei dynamischen Inhalten passiert es häufig, dass Copilot die „Code-

View“ (die nicht nutzerfreundliche Version eines dynamischen Inhaltes) direkt in ein Benutzeroberflächen-Feld schreibt. Diese Art der Eingabe ist von dem Konfigurations-Feld nicht verarbeitbar. Der dynamische Content (Name des Files), der im E-Mail-Body ausgegeben werden soll, wird zum Beispiel, wie in Abbildung 11 gezeigt, in der manuellen Benutzeransicht von einer Liste im Dropdown ausgewählt.



Abbildung 11: Auswahl dynamischer Inhalte

Diese Auswahl sieht in der Code-Ansicht wie folgt aus: „@{triggerBody()}?[{Name}]“.

Wenn Copilot versucht, diese Zeichenkette direkt in das Benutzeroberflächen-Feld einzufügen, führt das zu einem Fehler. Im schlimmsten Fall muss dann manuell eingegriffen werden.

4.2 Use-Case 1: Benachrichtigung bei Datei-Upload in SharePoint

Manuelle Erstellung

- Dauer der Erstellung: **18 Minuten**
- Durchschnittliche Durchlaufzeit: **[335984 ms; 25570ms]**
- Fehlerpunkte: **-1**

Der Fehlerpunkt bei der manuellen Erstellung entstand durch eine nicht erfolgreiche Ausführung des Workflows, bei der ein falscher dynamischer Content verwendet wurde.

Erstellung mit Microsoft Copilot

- Dauer der Erstellung: **62 Minuten**
- Durchschnittliche Durchlaufzeit: **[335972 ms; 25669 ms]**
- Fehlerpunkte: **-3**

Die Fehlerpunkte der KI-gestützten Erstellung ergaben sich aus einem leichten und einem mittelschweren Fehler. Der leichte Fehler wurde gewertet, weil die Mail zwar gesendet wurde, aber keine Daten enthielt. Mit dem „Flow-Builder-Chat“ wurde versucht die dynamischen Contents einzufügen. Dies zeigte aber keinen Erfolg und verbrauchte einiges an Zeit, was den großen Unterschied in der Erstellungsdauer unterstreicht. Copilot hatte Schwierigkeiten den dynamischen Content im Mail-Body zu setzen, was ein manuelles Eingreifen unumgänglich machte. Da der Flow aber grundsätzlich ausführbar war, wurde hier nur ein mittelschwerer Fehler gerechnet.

4.3 Use-Case 2: Extraktion von Tasks aus eingehenden E-Mails

Manuelle Erstellung

- Dauer der Erstellung: **63 Minuten**
- Durchschnittliche Durchlaufzeit: **[10314 ms; 3066ms]**
- Fehlerpunkte: **-3**

Die 3 Fehlerpunkte der manuellen Erstellung setzen sich aus drei leichten Fehlern zusammen. Diese leichten Fehler sind auf die Schwierigkeiten mit der „Parse-JSON“ Aktion zurückzuführen. Durch drei nicht erfolgreiche Ausführungen des Workflows entstanden die Fehlerpunkte.

Erstellung mit Microsoft Copilot

- Dauer der Erstellung: **71 Minuten**
- Durchschnittliche Durchlaufzeit: **[7700 ms; 4300 ms;]**
- Fehlerpunkte: **-13**

Leichte Fehler	Mittelschwere Fehler	Schwere Fehler
2	1	3

Tabelle 1: Fehlerpunktezusammensetzung Use-Case 2

Die hohe Anzahl an Fehlerpunkten lässt sich bei diesem Workflow auf die Verwendung von speziellen Aktionen zurückführen. Für diesen Use-Case wurde eine KI-Aktion verwendet, in welcher man einen eigenen Prompt für die Verarbeitung von Daten definieren kann. Die drei schweren Fehler sind jeweils auf entweder die Auswahl der Aktion, die Konfiguration der Aktion oder eine fehlende Logik zurückzuführen. Den ersten schweren Fehler stellt die Auswahl der KI-Aktion dar, diese konnte unmöglich von Copilot ausgewählt werden. Es wurden immer alternative Aktionen hinzugefügt, die aber die benötigte Funktionalität nicht abdeckten. Der zweite schwere Fehler entstand bei der Konfiguration der Aktionen, da die KI-Aktion nicht befüllt werden konnte. Es war nicht möglich einen ausführbaren Workflow mit Copilot zu generieren. Der dritte und letzte schwere Fehler entstand durch die fehlende Logik. Ohne die „Parse-JSON“ Aktion, konnten in dem Workflow die Daten nicht weiterverarbeitet werden. Diese Aktion wurde in keinem einzigen Versuch von Copilot erstellt, somit musste manuell nachgeholfen werden. Einer der zwei mittelschweren Fehler kommen ebenso aus dem „Parse-JSON“ da der „Flow-Builder-Chat“ eine falsche JSON Struktur erstellt hatte, mit der der Flow zwar ausführbar war, aber kein Ergebnis lieferte. Der zweite mittelschwere Fehler kam aus der KI-Aktion, welche im Gegensatz zur manuellen Version Probleme hatte, die

Tasks zu identifizieren und somit kein erfolgreicher Durchlauf möglich war. Die zwei leichten Fehler kamen aufgrund der Ausgabe der Daten zustande, dabei wurde das Mail nicht richtig formatiert und Daten wurden falsch dargestellt.

4.4 Use-Case 3: Zusammenfassung neuer SharePoint-News per E-Mail

Manuelle Erstellung

- Dauer der Erstellung: **28 Minuten**
- Durchschnittliche Durchlaufzeit: **[17267 ms; 13933 ms]**
- Fehlerpunkte: **0**

Erstellung mit Microsoft Copilot

- Dauer der Erstellung: **101 Minuten**
- Durchschnittliche Durchlaufzeit: **[15950 ms; 13250 ms]**
- Fehlerpunkte: **-14**

Leichte Fehler	Mittelschwere Fehler	Schwere Fehler
2	-	4

Tabelle 2: Fehlerpunktezusammensetzung Use-Case 3

Der Copilot generierten Workflows für Use-Case 4 ist der Workflow mit den meisten Fehlerpunkten. Bei diesem Workflow lag das Problem wie bei Use-Case 3 zu einem großen Teil an der KI-Aktion. Erstellung und Konfiguration der KI-Aktion waren hier ebenso nicht möglich wie in Use-Case 3 was zu den ersten zwei schweren Fehlern führte. Der dritte Fehler entstand durch die falsche Konfiguration der „HTTP Request“ Aktion. Diese ist verantwortlich für das Abfragen der neuen SharePoint-News. Copilot war es nicht möglich die Aktion mit der richtigen API-Abfrage zu konfigurieren, geschweige denn einen dynamischen Content als Parameter in der API-Abfrage zu verwenden. Der letzte schwere Fehler entstand durch das nicht anlegen von Variablen. Für die Verarbeitung der Daten muss eine Variable initialisiert und gesetzt werden. Auch dieser Teil konnte von Copilot nicht erstellt werden und führte somit zu einem nicht ausführbaren Workflow. Die zwei leichten Fehler, bezogen sich auch wie in Use-Case 3 auf die Ausgabe der Daten. Dabei wurden Felder wie, Betreff in der Mail ausgelassen oder mit falschen Werten befüllt und die Formatierung der Ausgabedaten war nicht wie erwünscht. Durch die unzähligen Versuche, die KI-Aktion und den HTTP-Request von Copilot erstellen zu lassen stieg die Erstellungsdauer sehr stark an.

4.5 Use-Case 4: Automatisierte Genehmigung für Zugriff auf Teams-Team

Manuelle Erstellung

- Dauer der Erstellung: **19 Minuten**
- Durchschnittliche Durchlaufzeit: **[29511 ms; 12889 ms]**
- Fehlerpunkte: **0**

Erstellung mit Microsoft Copilot

- Dauer der Erstellung: **44 Minuten**
- Durchschnittliche Durchlaufzeit: **[62207 ms; 36993 ms]**
- Fehlerpunkte: **-5**

In Use-Case 4 errechnen sich die Fehlerpunkte aus einem schweren Fehler und einem mittelschweren Fehler. Der mittelschwere Fehler ist dabei wieder auf eine fehlende Konfiguration zurückzuführen. Die Aktion „Hinzufügen eines Mitglieds zu einem Team“ braucht als Konfiguration ein Teams Team, zu welchem der User hinzugefügt werden soll. Dabei ist das Team aus einer Dropdown-Liste auszuwählen. Copilot ist es nicht möglich einen Wert aus einer Drop-Down-Liste auszuwählen, daher musste der Wert im Prompt nicht als Anzeigename angegeben werden, sondern als GUID. Der schwere Fehler bezieht sich ebenso auf dieselbe Aktion. Dabei konnte der dynamische Content für den User, der hinzugefügt werden soll, nicht gesetzt werden und somit der Workflow in einen Fehler erzeugte.

4.6 Use-Case 5: Automatische Teams-Zugriffsberechtigung für neue IT-Mitarbeiter

Manuelle Erstellung

- Dauer der Erstellung: **23 Minuten**
- Durchschnittliche Durchlaufzeit: **[1020 ms; 715 ms]**
- Fehlerpunkte: **-1**

Der leichte Fehler in der manuellen Erstellung ist auf die falsche Verwendung einer Aktion zurückzuführen. Dabei wurde für die Abfrage der Azure-Gruppe eine falsche Aktion verwendet, welche eine falsche Ausgabe lieferte.

KI-gestützte Erstellung mit Microsoft Copilot

- Dauer der Erstellung: **86 Minuten**
- Durchschnittliche Durchlaufzeit: **[1385ms; 654ms]**
- Fehlerpunkte: **-12**

Leichte Fehler	Mittelschwere Fehler	Schwere Fehler
-	3	2

Tabelle 3: Fehlerpunktezusammensetzung Use-Case 5

Die schweren Fehler sind beide auf die „Bedingung“ Aktion zurückzuführen. Der erste schwere Fehler war das Copilot nicht erkannte das eine Bedingung (If-Abfrage) notwendig ist. Copilot hat sich vehement gewehrt die Abfrage hinzuzufügen. Der zweite schwere Fehler entstand durch die Konfiguration der Abfrage. Es war nicht möglich aus natürlicher Sprache die Bedingung zu konfigurieren. Einer der zwei mittelschweren Fehler ist gleich wie in Use-Case 4 bei dem das Teams Team nicht per Dropdown ausgewählt werden konnte. Der zweite mittelschwere Fehler entstand beim Konfigurieren der „Hinzufügen eines Mitglieds zu einer Team“ Aktion. Dabei wurde der Flow-Builder-Chat verwendet, welcher es nicht schaffte in die bestehende Aktion einen dynamischen Content einzufügen. Hier wurden immer wieder neue Aktionen außerhalb der Bedingung hinzugefügt. Wie bereits im Use-Case 3 erforderten auch hier die Versuche, das Problem mit Copilot zu beheben, viel Zeit.

Diese Ergebnisse geben erste Antworten auf die formulierte Forschungsfrage und zeigen konkrete Stärken und Schwächen auf. Im anschließenden Kapitel werden diese Ergebnisse interpretiert und in den Kontext der Literatur sowie praktischer Implikationen eingeordnet

5. Diskussion und Beantwortung der Forschungsfrage

5.1 Zusammenfassung der Ergebnisse

Zur besseren Vergleichbarkeit der erhobenen Daten wurde eine tabellarische Übersicht erstellt, welche die zentralen Metriken, Erstellungsdauer, durchschnittliche Durchlaufzeit und Fehlerpunkte für alle fünf Use-Cases sowohl für die manuelle als auch für die Copilot-gestützte Erstellung gegenüberstellt. Diese systematische Gegenüberstellung ermöglicht es, Muster in der Leistungsfähigkeit beider Ansätze zu erkennen und dient als Grundlage für die anschließende Bewertung der Hypothese.

USE-CASE	MANUELL ODER COPILOT	GESAMTDAUER DER ERSTELLUNG	Ø DURCHLAUFZEIT MIT STANDARADABWEICHUNG	FEHLERPUNKTE GESAMT
USE-CASE 1	Manuell	18 min	[335984 ms; 25570 ms]	-1
USE-CASE 1	Copilot	62 min	[335972 ms; 25669 ms]	-3
USE-CASE 2	Manuell	63 min	[10314 ms; 3066 ms]	-3
USE-CASE 2	Copilot	71 min	[7700 ms; 4300 ms;]	-13
USE-CASE 3	Manuell	28 min	[17267 ms; 13933 ms]	0
USE-CASE 3	Copilot	101 min	[15950 ms; 13250 ms]	-14
USE-CASE 4	Manuell	19 min	[29511 ms; 12889 ms]	0
USE-CASE 4	Copilot	44 min	[62207 ms; 36993 ms]	-5
USE-CASE 5	Manuell	23 min	[1020 ms; 715 ms]	-1
USE-CASE 5	Copilot	86 min	[1385 ms; 654 ms]	-12

Tabelle 4: Zusammenfassung der Ergebnisse

Aus der Übersicht lassen sich mehrere zentrale Beobachtungen ableiten. Die manuelle Erstellung der Workflows war in allen Fällen mit einer geringeren Erstellungsdauer verbunden als die KI-gestützte Variante. Diese Metrik zeigt, dass im Rahmen der Bachelorarbeit die manuelle Erstellung bei allen Use-Cases schneller erfolgte als die durch Copilot-gestützte Erstellung. Daraus könnte man ableiten, dass Microsoft Copilot, trotz seiner Funktion als unterstützendes Tool, aktuell noch einen höheren Einarbeitungs- und Korrekturaufwand verursacht. Gleichzeitig zeigen die Durchlaufzeiten der Copilot-generierten Workflows in einigen Fällen Vorteile. Sie sind im Betrieb tendenziell schneller, was auf eine technisch effizientere Ausführung hinweisen könnte. Da während der Erstellung der Workflows Unterschiede erkannt wurden, sollte die Durchlaufzeit besonders berücksichtigt werden. Die Durchlaufzeit eines Flows ist nie konstant, sie variiert zwischen den einzelnen Durchläufen. Potenziell kann das am

Aufbau von Power Automate als Cloud Software liegen. Da Cloud-Systeme Ressourcen-Schwankungen ausgesetzt sind, kann sich somit die Ausführung des gleichen Flows länger oder kürzer sein, je nach freien Ressourcen. Besonders markant ist der Unterschied bei den Fehlerpunkten. Während die manuelle Erstellung in den meisten Fällen fehlerfrei oder mit leichten bis maximal mittelschweren Fehlern durchgeführt werden konnte, wiesen die durch Copilot erzeugten Workflows eine wesentlich höhere Fehleranfälligkeit auf. Dieser Unterschied verdeutlicht die Herausforderungen, welche beim Interpretieren von Prompts, bei der fehlerfreien Logikabbildung sowie bei der Verwendung dynamischer Inhalte bestehen.

Use-Case 2,3 und 5 stechen im direkten Vergleich besonders heraus. Bei allen traten bei der Copilot-Version mehrfach mittelschwere bis schwere Fehler auf, die eine erfolgreiche Ausführung zunächst verhinderten. Hingegen dazu verliefen die manuellen Varianten dieser Use-Cases ohne oder mit minimalen Problemen. Daraus lässt sich ableiten, dass Copilot derzeit insbesondere bei logisch komplexeren Automatisierungsprozessen noch nicht durchgängig verlässlich funktioniert.

Im Detail zeigten sich während der praktischen Erstellung der Workflows mittels Microsoft Copilot spezifische Schwierigkeiten. Bei einfachen Workflows (Use-Case 1) offenbarte Copilot bereits Probleme bei der korrekten Integration dynamischer Inhalte, was zu manuellen Anpassungen führte. Bei komplexeren Anforderungen (Use-Case 2 und 3) erwies sich Copilot teilweise als ungeeignet, da die Erkennung der passenden Aktionen und die korrekte technische Umsetzung nicht möglich war. Besonders deutlich wurden diese Grenzen beim Erstellen von Workflows, die eine komplexere technische Logik (Use-Case 5) oder eine Textzusammenfassung (Use-Case 3) erfordern. Hier wurden technische Anforderungen vielfach missinterpretiert, und es war nahezu unmöglich, zufriedenstellende Ergebnisse, ohne umfassende manuelle Korrekturen zu erreichen. Auch einfache Interaktionen wie Dropdown-Felder wurden nicht immer zuverlässig konfiguriert (Use-Case 4). Dies verdeutlicht, dass Copilot in der aktuellen Version noch erhebliche Schwächen aufweist, insbesondere bei technisch anspruchsvollen oder dynamischen Automatisierungen.

5.2 Beantwortung der Forschungsfrage im Kontext der Hypothese

Die ursprüngliche Hypothese dieser Arbeit lautet:

„Die von Microsoft Copilot erstellten Microsoft Power Automate Workflows sind effizienter und schneller, weisen jedoch eine höhere Fehleranfälligkeit auf im Vergleich zu manuell erstellten Workflows.“

Nach den Ergebnissen der durchgeführten Fallstudie zeigt sich, dass die Hypothese hinsichtlich der Schnelligkeit teilweise bestätigt, hinsichtlich der Fehleranfälligkeit übererfüllt und hinsichtlich der Effizienz widerlegt wird.

5.2.1 Effizienz (Dauer der Workflow-Erstellung)

Entgegen der ursprünglichen Annahme waren die Copilot-generierten Workflows hinsichtlich der Effizienz bei der Erstellung nicht überlegen. Im Gegenteil, in allen untersuchten Fällen war die Erstellung der Workflows mithilfe von Microsoft Copilot zeitintensiver als die manuelle Erstellung. Vor allem in Use-Case 3 war die Erstellung durch Copilot mit 101 Minuten deutlich zeitaufwendiger als die manuelle Umsetzung mit nur 28 Minuten, was erneut den höheren Korrektur- und Anpassungsaufwand bestätigt. Dies steht im Widerspruch zur Hypothese, welche eine höhere Effizienz der Copilot-generierten Workflows vorausgesetzt hatte. Ursprünglich wurde angenommen, dass die KI-Unterstützung den Erstellungsprozess beschleunigen würde, die Praxis zeigt jedoch klare Grenzen der aktuellen KI-Technologie auf. Daher sollten Unternehmen ihre Erwartungen an kurzfristige Produktivitätssteigerungen durch den Einsatz von Microsoft Copilot realistisch einschätzen und berücksichtigen, dass besonders bei komplexeren Szenarien erheblicher zusätzlicher Aufwand entstehen kann.

5.2.2 Schnelligkeit (Durchlaufzeit der Workflows)

Die Hypothese hinsichtlich einer verbesserten Durchlaufzeit konnte nur teilweise bestätigt werden. In drei von fünf Use-Cases (1, 2 und 3) zeigten die von Copilot generierten Workflows eine geringere durchschnittliche Durchlaufzeit im Vergleich zu den manuell erstellten. Besonders deutlich war dies in Use-Case 2. In den Use-Cases 4 und 5 hingegen war die Durchlaufzeit mit Copilot sogar spürbar länger. Diese gemischten Ergebnisse relativieren den erwarteten Vorteil und zeigen, dass Copilot zwar in bestimmten Szenarien operative Geschwindigkeitsvorteile bringen kann, dies aber nicht allgemein zutrifft. Diese Unterschiede verdeutlichen, dass die Vorteile von Copilot stark vom spezifischen Anwendungskontext abhängen. Während einfachere Workflows von einer schnelleren Ausführung profitieren können, ist dies bei komplexeren oder technisch anspruchsvollen Abläufen nicht durchgängig der Fall. Unternehmen sollten diese kontextspezifischen Unterschiede bei der Entscheidung über den Einsatz von Copilot berücksichtigen.

5.2.3 Fehleranfälligkeit

Die erwartete höhere Fehleranfälligkeit der Copilot-generierten Workflows bestätigte sich ebenfalls in den Ergebnissen. In allen untersuchten Fällen traten bei der KI-gestützten Erstellung öfter und auch schwerwiegendere Fehler auf, während die manuell

erstellten Workflows kaum oder keine Fehler aufwiesen. Besonders deutlich ist das in Use-Case 3 zu erkennen, bei dem der Copilot-Workflow mit -14 Fehlerpunkten auffiel. Das ist ein Hinweis auf Schwierigkeiten bei der inhaltlichen Interpretation, der dynamischen Datenzuordnung und der richtigen Verwendung und Konfiguration von Aktionen in Power Automate.

5.2.4 Beantwortung der Forschungsfrage

Aus den Ergebnissen lässt sich schließen, dass Microsoft Copilot in der aktuellen Form bei der Workflow-Automatisierung zwar Vorteile bezüglich der Durchlaufzeit in der operativen Ausführung bieten kann, zugleich jedoch Schwächen hinsichtlich der initialen Erstellungseffizienz sowie der Stabilität und Fehleranfälligkeit aufweist. Besonders im Hinblick auf den Trend des „Citizen Developments“, bei dem Fachanwender eigenständig Prozesse automatisieren, ist die derzeitige Situation kritisch zu betrachten. Unternehmen sollten daher sicherstellen, dass Citizen Developer entweder ausreichend geschult werden, um mit den Fehlern und Nacharbeiten umgehen zu können, oder die Einführung von Copilot durch erfahrene IT-Fachkräfte begleitet wird. Gerade weniger technisch versierte Nutzer könnten durch die bestehenden Fehleranfälligkeiten und die Notwendigkeit der manuellen Nachkorrekturen schnell überfordert sein, was dem eigentlichen Mehrwert von Low-Code-Plattformen entgegenwirkt.

Gleichzeitig bringt Microsoft Copilot ein großes Potenzial mit, insbesondere im Bereich Power Automate für Personal Use. Für individuelle, kleinere Automatisierungen, bei denen Nutzer ihre persönlichen Arbeitsprozesse mit einfachen KI-generierten Workflows unterstützen möchten, könnte Copilot bereits heute eine wertvolle Hilfe sein, vorausgesetzt, die Komplexität der Aufgaben bleibt gering und die Nutzer bringen ein gewisses Grundverständnis für Workflow-Logik mit.

Insgesamt zeigt sich, dass Copilot durchaus das Potenzial hat, Anwender langfristig zu entlasten und den Citizen-Development-Ansatz zu stärken, aktuell jedoch zurzeit noch keine verlässliche Alternative zu manuellen, professionell erstellten Workflows darstellt.

5.3 Vergleich mit der Literatur

Die Ergebnisse der durchgeführten experimentellen Fallstudie bieten interessante Anhaltspunkte für den Vergleich mit bisherigen Erkenntnissen aus der Fachliteratur.

5.3.1 Übereinstimmungen mit bestehenden Erkenntnissen

Die in dieser Arbeit beobachteten Vorteile von Microsoft Copilot, insbesondere die teilweise schnellere Durchlaufzeit bei Workflow-Ausführungen, decken sich mit den Erkenntnissen von Simao & Shaddoc (2023). Laut ihrer Studie ermöglichen KI-Systeme eine raschere Implementierung und eine zügigere Ausführung von Workflows. Dieser

Punkt wurde in Use-Case 1 und Use-Case 4 ebenfalls bestätigt, wobei sich die Bestätigung lediglich auf die zügigere Ausführung beziehen lässt.

Die Ergebnisse aus Use-Case 5 bestätigen die von Afrin et al. (2025) beschriebenen Risiken KI-gestützter Systeme, insbesondere deren Fehleranfälligkeit bei unvorhergesehenen oder komplexen Abläufen, auch wenn sie grundsätzlich Potenzial für den Einsatz bei datenintensiven Aufgaben zeigen. Besonders kritisch waren mittelschwere und schwere Fehler, die entweder die korrekte Workflow-Ausführung vollständig verhinderten oder erhebliche manuelle Korrekturen notwendig machten. Diese Risiken entsprechen den in der Literatur beschriebenen Herausforderungen nach Afrin et al., (2025). Unternehmen sollten deshalb sorgfältig prüfen, für welche Anwendungsfälle KI-Tools eingesetzt werden, und gegebenenfalls zusätzliche Qualitätskontrollen einführen.

5.3.2 Unterschiede und Ergänzungen zur Literatur

Interessant ist der klare Unterschied zwischen Theorie und Praxis bezüglich der Effizienz bei der Erstellung von Workflows. Während Afrin et al. (2025) und Pandey et al. (2024) insbesondere Zeitersparnisse und Effizienzsteigerung durch KI-Unterstützung in den Vordergrund stellen, zeigte sich in dieser praktischen Untersuchung deutlich, dass der zusätzliche Aufwand für Nachkorrekturen die initialen theoretischen Vorteile überwiegt. Dies deutet darauf hin, dass die praktischen Herausforderungen, insbesondere bei der Interaktion zwischen Benutzer und KI, möglicherweise bisher unterschätzt wurden.

5.3.3 Implikationen aus dem Literaturvergleich

Die Resultate dieser Bachelorarbeit erweitern die bestehende Literatur um eine gezielte und praxisnahe Perspektive auf die Schnelligkeit, Effizienz und Fehleranfälligkeit von KI-gestützten Automatisierungssystemen. Sie bestätigen einerseits die grundsätzlichen Vorteile von KI-Tools in der Workflow-Automatisierung, zeigen jedoch auch deutlich auf, dass die Technologie in ihrem derzeitigen Stand noch auf einige Herausforderungen trifft, die in der Theorie oft weniger prominent erscheinen. Insgesamt zeigen die Ergebnisse, dass KI-gestützte Automatisierungssysteme zwar über vielversprechendes Potenzial verfügen, in ihrem aktuellen Entwicklungsstand jedoch insbesondere im Enterprise-Kontext noch nicht durchgängig praxistauglich sind.

6. Schlussfolgerung und Ausblick

6.1 Schlussbetrachtung der gesamten Bachelorarbeit

Diese Bachelorarbeit beschäftigte sich mit der Frage, ob Microsoft Copilot in der Lage ist, funktionale und effiziente Workflows in Microsoft Power Automate zu erstellen und wie sich diese im Vergleich zu manuell entwickelten Workflows verhalten. Dazu wurde im Rahmen einer experimentellen Fallstudie ein direkter Vergleich anhand von fünf praxisnahen Use-Cases durchgeführt. Gemessen wurden die Durchlaufzeit, die Fehleranfälligkeit und zusätzlich die Erstellungsdauer der jeweiligen Workflows. Die Ergebnisse zeigten, dass Copilot-gestützte Workflows zwar potenziell zeitsparender in der Ausführung sind, jedoch in der Erstellung aufwendiger und fehleranfälliger als manuell erstellte Automatisierungen.

6.2 Fazit aus dem empirischen Teil

Die empirische Auseinandersetzung mit dem Thema bestätigte, dass Microsoft Copilot in seiner aktuellen Version in Power Automate keine vollständig zuverlässige Alternative zur manuellen Erstellung von Workflows darstellt. Die Effizienz bei der Erstellung war durchwegs geringer als bei den manuellen Gegenstücken. Zwar konnten Copilot-Workflows teilweise in der Ausführung eine etwas schnellere Durchlaufzeit erreichen, allerdings war der Unterschied meist eher gering. Deutlich wurde die erhöhte Fehleranfälligkeit insbesondere bei komplexeren Anforderungen oder mehrstufigen Logiken.

Positiv überrascht hat die Fähigkeit von Copilot, eine funktionale Grundstruktur für Workflows zu liefern. Für erfahrenere Nutzer kann dies eine hilfreiche Ausgangsbasis sein, um nicht jedes Projekt von Grund auf neu zu beginnen. Enttäuschend war hingegen die Instabilität bei der Konfiguration der Aktionen im Flow-Builder sowie die eingeschränkte Fähigkeit von Copilot, mit unklaren oder komplexen Anforderungen umzugehen. Gerade in diesen Fällen führten automatische Änderungen oder ungewollte Löschungen zu einem erhöhten Arbeitsaufwand.

6.3 Implikation für die Praxis und zukünftige Forschung

Die Ergebnisse dieser Arbeit zeigen, dass Unternehmen den Einsatz von Microsoft Power Automate, insbesondere in Kombination mit Microsoft Copilot, sorgfältig planen sollten. Zwar bietet das Konzept des „Citizen Developments“ grundsätzlich großes Potenzial, um IT-Abteilungen zu entlasten. Allerdings erfordert dies benutzerfreundliche und stabile Automatisierungstools, die auch Mitarbeitende ohne tiefgehende technische Kenntnisse zuverlässig nutzen können. Aktuell erfüllt Copilot diese Anforderung lediglich bei einfachen Automatisierungsszenarien vollständig, während komplexere

Anforderungen schnell die Grenzen der KI aufzeigen. Dies kann in der Praxis zu Frustration, zusätzlichen Supportanfragen und einer Belastung der IT-Abteilung führen, die sich mit fehlerhaften oder unvollständigen Workflows auseinandersetzen muss.

Um diese Herausforderungen zu bewältigen, sollten Unternehmen klare Governance-Strukturen etablieren, optional unterstützende Schulungsangebote bereitstellen und das Bewusstsein für die Verantwortung beim Erstellen automatisierter Workflows schärfen. Insbesondere bei Unternehmen mit älteren Belegschaften könnte ein schrittweiser Ansatz hilfreich sein, bei dem zunächst Pilotgruppen Erfahrungen sammeln und ihr Wissen anschließend unternehmensweit weitergeben. Dieser Ansatz minimiert Risiken und erhöht gleichzeitig die Akzeptanz und Effektivität der eingeführten Automatisierungslösungen.

Für zukünftige Forschung ergeben sich aus diesen Erkenntnissen mehrere vielversprechende Ansatzpunkte. Zum einen könnten verbesserte KI-Modelle, insbesondere durch Methoden des Prompt-Engineerings und durch Optimierungen der Interaktionsschnittstellen, genauer untersucht werden, um die Fehleranfälligkeit bei der Workflow-Erstellung signifikant zu reduzieren. Darüber hinaus wäre es aufschlussreich, verschiedene KI-Systeme wie, ChatGPT, Google Gemini oder Anthropic Claude, gezielt im Kontext der Workflow-Automatisierung zu vergleichen. Ein solcher systematischer Vergleich könnte dazu beitragen, die jeweiligen Stärken und Schwächen von den KI-Systemen besser zu verstehen und gezielter Einsatzempfehlungen für verschiedene Anwendungsszenarien abzuleiten.

Darüber hinaus empfehlen sich verstärkt praxisnahe Untersuchungen, um realistische Erwartungshaltungen und typische Problembereiche beim Einsatz von KI-gestützten Automatisierungswerkzeugen frühzeitig zu erkennen. Unternehmen sollten zunächst einfache und repetitive Aufgaben automatisieren, um Erfahrungen zu sammeln und die damit verbundenen Risiken zu minimieren. Komplexere Prozesse, bei denen langfristige Stabilität und geringe Fehleranfälligkeit essenziell sind, sollten aktuell bevorzugt manuell realisiert werden. Hingegen könnten Anwendungsbereiche, in denen Flexibilität und Geschwindigkeit besonders gefragt sind, durchaus von Copilot-generierten Workflows profitieren.

Literaturverzeichnis

- Afrin, S., Roksana, S., & Akram, R. (2025). AI-Enhanced Robotic Process Automation: A Review of Intelligent Automation Innovations. *IEEE Access*, 13, 173–197. IEEE Access. <https://doi.org/10.1109/ACCESS.2024.3513279>
- Barke, S., James, M. B., & Polikarpova, N. (2022). *Grounded Copilot: How Programmers Interact with Code-Generating Models* (No. arXiv:2206.15000). arXiv. <https://doi.org/10.48550/arXiv.2206.15000>
- camillepack. (2024, November 19). *Microsoft 365 Copilot Overview*. <https://learn.microsoft.com/en-us/copilot/microsoft-365/microsoft-365-copilot-overview>
- Ekin, S. (2023). *Prompt Engineering For ChatGPT: A Quick Guide To Techniques, Tips, And Best Practices*. <https://doi.org/10.36227/techrxiv.22683919.v2>
- Guilmette, A. (2020). *Workflow automation with microsoft power automate: Achieve digital transformation through business automation with minimal coding*. Packt Publishing.
- Hofmann, P., Samp, C., & Urbach, N. (2020). Robotic process automation. *Electronic Markets*, 30(1), 99–106. <https://doi.org/10.1007/s12525-019-00365-8>
- Hyun, Y., Lee, D., Chae, U., Ko, J., & Lee, J. (2021). Improvement of Business Productivity by Applying Robotic Process Automation. *Applied Sciences*, 11(22), Article 22. <https://doi.org/10.3390/app112210656>
- Khan, A. (2024). *Introducing Microsoft Copilot for Managers: Enhance Your Team's Productivity and Creativity with Generative AI-Powered Assistant*. Apress. <https://doi.org/10.1007/979-8-8688-0419-9>
- Koch, O., & Wildner, S. (2020). Intelligent Robotic Process Automation: Konzeption eines Ordnungsrahmens zur Nutzung künstlicher Intelligenz für die Prozessautomatisierung. In R. Buchkremer, T. Heupel, & O. Koch (Hrsg.), *Künstliche Intelligenz in Wirtschaft & Gesellschaft* (S. 211–230). Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-29550-9_12
- Liu, Y., He, H., Han, T., Zhang, X., Liu, M., Tian, J., Zhang, Y., Wang, J., Gao, X., Zhong, T., Pan, Y., Xu, S., Wu, Z., Liu, Z., Zhang, X., Zhang, S., Hu, X., Zhang, T., Qiang, N., ... Ge, B. (2025). Understanding LLMs: A comprehensive overview from training to inference. *Neurocomputing*, 620, 129190. <https://doi.org/10.1016/j.neucom.2024.129190>
- Pandy, G., Jayaram, V., Krishnappa, M. S., Ingole, B. S., Ganeeb, K. K., & Joseph, S. (2024). Advancements in Robotics Process Automation: A Novel Model with Enhanced Empirical Validation and Theoretical Insights. *European Journal of Computer Science and Information Technology*, 12(5), 64–73. <https://doi.org/10.37745/ejcsit.2013/vol12n56473>

- Pearson, M., Knight, B., Knight, D., & Quintana, M. (2020). *Pro Microsoft Power Platform: Solution Building for the Citizen Developer*. Apress. <https://doi.org/10.1007/978-1-4842-6008-1>
- Ribeiro, J., Lima, R., Eckhardt, T., & Paiva, S. (2021). Robotic Process Automation and Artificial Intelligence in Industry 4.0 – A Literature review. *Procedia Computer Science*, 181, 51–58. <https://doi.org/10.1016/j.procs.2021.01.104>
- Sibalija, T., Jovanović, S., & Đurić, J. (2019). *ROBOTIC PROCESS AUTOMATION: OVERVIEW AND OPPORTUNITIES*.
- Simao, T., & Shaddoc, A. (2023). Strategic Integration of AI and RPA for Enhanced Organizational Efficiency. *International Journal of Advanced Engineering Technologies and Innovations*, 1(04), Article 04.
- Stratton, J. (2024). An Introduction to Microsoft Copilot. In J. Stratton, *Copilot for Microsoft 365* (S. 19–35). Apress. https://doi.org/10.1007/979-8-8688-0447-2_2
- v-aangie. (2024, September 4). *Copilot in Power Automate*. <https://learn.microsoft.com/en-us/power-automate/copilot-overview>

Abbildungsverzeichnis

Abbildung 1: Copilot Prompt Verarbeitung (camillepack, 2024)	6
Abbildung 2: Copilot Erstellung mit Use-Case Definition	22
Abbildung 3: Copilot generierter Workflow mit Use-Case Definition	22
Abbildung 4: Unkonfigurierter Workflow	23
Abbildung 5: Copilot Erstellung mit eigenem Prompt	24
Abbildung 6: Copilot generierter Workflow mit eigenem Prompt	24
Abbildung 7: Teils Konfigurierter Workflow	25
Abbildung 8: Aktions-Konfiguration mit Use-Case Definition	25
Abbildung 9: Aktions-Konfiguration mit eigenem Prompt	25
Abbildung 10: Copilot im Flow-Builder	26
Abbildung 11: Auswahl dynamischer Inhalte	27

Tabellenverzeichnis

Tabelle 1: Fehlerpunktezusammensetzung Use-Case 2	28
Tabelle 2: Fehlerpunktezusammensetzung Use-Case 3	29
Tabelle 3: Fehlerpunktezusammensetzung Use-Case 5	31
Tabelle 4: Zusammenfassung der Ergebnisse	32