Editorial

Christian Müller

Norbert Ketterer

Frank Herrmann

Liebe Leserinnen und Leser,

vor Ihnen liegt nunmehr die bereits einundzwanzigste Ausgabe des E-Journals Anwendungen und Konzepte in der Wirtschaftsinformatik (AKWI) – wir hoffen, dass wir Ihnen wieder eine Reihe von spannenden Artikeln aus dem Umfeld der Wirtschaftsinformatik zusammenstellen konnten. Wir möchten auch noch einmal darauf hinweisen, dass die regulären Artikel alle durch einen komplett anonymisierten Review-Prozess laufen, in dem zwei Gutachter und ein Redakteur/ Herausgeber den Artikel begleiten.

Wie in den bisherigen Ausgaben decken die Beiträge ein breites Spektrum klassischer und aktueller Themen der Wirtschaftsinformatik ab. Die vorliegende Ausgabe umfasst zwölf Beiträge, darunter drei Kurzdarstellungen von Abschlussarbeiten. Inhaltlich reichen die Arbeiten von grundlegenden Konzepten über praxisorientierte Projekt- und Systembeiträge bis hin zu aktuellen Trendthemen im Kontext Künstlicher Intelligenz.

Im Bereich der Grundlagen wird die Bedeutung von Checklisten in sicherheitskritischen Domänen beleuchtet. Der Beitrag Checklist Application in Aviation and Healthcare zeigt, wie Checklisten als strukturierende und komplexitätsreduzierende Werkzeuge in Luftfahrt und Medizin eingesetzt werden und welche Voraussetzungen für ihre Wirksamkeit entscheidend sind.

Ein Schwerpunkt der Ausgabe liegt im Bereich Praxis und Anwendungssysteme. Am Beispiel der HENRICHSEN4s wird gezeigt, wie eine automatisierte Projektrenditenberechnung mit Power BI umgesetzt werden kann. Die Analyse zu Möglichkeiten und Grenzen KI-unterstützter Softwareentwicklung im SAP-Umfeld definiert Bedarfe an Codegenerierungstools und vergleicht GitHub Copilot, Tabnine und Codeium, einschließlich relevanter Fragestellungen zu Daten und Nutzung. Architekturelle Aspekte adressiert der Beitrag zur Entwicklung einer integrierten Microservice-Architektur für modularisierte RPA-Prozesse, der monolithische Abläufe in wiederverwendbare, lose gekoppelte Services überführt und dies durch Orchestrierung, Katalogisierung sowie Experteninterviews absichert. Ergänzend wird in einem praxisorientierten Vergleich von SAP GUI und SAP Fiori herausgearbeitet, welche Vor- und Nachteile beide Frontends aufweisen und warum Fiori insbesondere für neu entwickelte Anwendungen sowie in Ausbildungskontexten Vorteile bieten kann.

Weitere Beiträge befassen sich mit betriebswirtschaftlichen, logistischen und politischen Aspekten der Digitalisierung. So wird ein Konzept zur Monetarisierung und dynamischen, zuschlagbasierten Preisermittlung für freie Software am Beispiel von OpenSlides entwickelt und in einer Testumgebung durchlaufen, um die Umsetzbarkeit in der Praxis zu validieren. Im Kontext der Intralogistik wird eine prioritätsregelbasierte Traffic-Management-Policy für die Koordination von AGV-Flotten in Konfliktzonen vorgestellt, die die Anpassungsfähigkeit und Skalierbarkeit gegenüber oft starren Koordinationsmethoden verbessern soll. Mit der Evaluation der zweiten Förderperiode der Digitalprämie Berlin wird zudem ein Beitrag präsentiert, der die Wirkung eines öffentlichen Förderprogramms für die Digitalisierung von KMU analysiert, Investitionsschwerpunkte wie IT-Sicherheit und Automatisierung herausstellt und Empfehlungen zur Vereinfachung sowie Standardisierung der Vergabe- und Evaluationsprozesse ableitet.

Im Bereich Trends steht die Weiterentwicklung des KI-gestützten Hochschul-Chatbots „Winfy“ im Vordergrund. Der Beitrag zeigt, wie mittels eines domänenspezifischen Transformers für Extractive Question Answering (EQA) Antworten gezielter aus Antwortblöcken extrahiert und damit die Antwortgenauigkeit erhöht werden können.

Die drei Kurzdarstellungen von Abschlussarbeiten ergänzen die Ausgabe um aktuelle Forschungsperspektiven: Eine Arbeit untersucht empirisch die Erkennung und ethische Bewertung KI-generierter Bilder. Eine weitere evaluiert Process Mining als Analyseinstrument für die Produktionslogistik eines Automobilherstellers. Die dritte ordnet Potenziale und Grenzen der SAP Business Technology Platform als Zukunftsoption für KMU mit SAP Business One ein, insbesondere mit Blick auf Kostenaspekte.

Über Ihr Interesse an der Zeitschrift freuen wir uns und wünschen Ihnen Freude bei der Lektüre.

Regensburg, Fulda, Luzern und Wildau, im Dezember 2025.

Frank Herrmann, Norbert Ketterer und Christian Müller

CHECKLIST APPLICATION IN AVIATION AND HEALTHCARE

Stefanie Eichstedt
Department of Industrial Engineering
University of Rome Tor Vergata
Via del Politecnico 1
00133 Rome, Italy
E-Mail: steffie.eichstedt@gmx.de

ABSTRACT

Aviation and healthcare are professional domains that are characterized by high safety requirements. Human failure is identified as the major cause for aviation accidents and medical complications. In both fields, checklists – that are applied in regular and abnormal situations – are reliable and capable tools to meet the high safety aspirations and to reduce or avoid mistakes and their partially severe consequences. The benefits of this rather simple method are exploited best when they are embedded in an overall safety culture of the organization. Checklist discipline, correct application and teamwork are key for their effective contribution to safety – be it in surgery or flight. Moreover, checklists support decision-making due to their capacity to reduce complexity and ambiguity. The structured, standardized proceeding is in particular valuable. This paper describes how checklists function generally and how their application in aviation spilled over to medicine and became also indispensable in daily business.

KEYWORDS

checklist, safety, aviation, medicine, decision-making

INTRODUCTION

The utilization of checklists as a tool for safety and quality management is widely established and appreciated in many professional domains. This holds also true for the fields of aviation and healthcare which clearly benefit from the application of specifically developed, purposeful checklists. Both branches have a variety of commonalities that are related to the responsible work in environments that are characterized by specific safety requirements, a significant workload under (time) pressure, the occurrence of distractions, indispensable team-coordination and considerable amounts of (changing) information. In order to meet the particular safety and quality standards in both fields, checklists are an effective support because of their capability to organize complexity, reduce ambiguity – without simplification – and to promote professional decision-making. Their successful implementation and diligent application are closely tied to certain preconditions. This relates among others to the demands regarding checklist design, the precise representation of the content and the indispensable acceptance of the users. This paper is intended to describe the methodology behind checklists and to demonstrate their contribution as a practical decision aid for medicine and aviation. In general, checklists are practical working aids in the form of concisely written documents[1] with a definite purpose orientation. Scope and envisioned context define the development of the concrete checklist. Their core function is being a memory aid that relieves the user cognitively and thereby prevents or reduces avoidable mistakes resulting from distracted attention and limited human capacities. They can help to structure complex processes or tasks, ensure and document completeness, consistency and compliance with due diligence obligations. Needless to say, they do not replace professional skills, judgment, knowledge or expertise at all. They rather represent reliable support, routine and structure in regular day-to-day work or in abnormal situations. Regardless of their practicality and simplicity of application, checklists have a multitude of advantages, that are summarized in the following table.

[1] Here, focus is put on the written/printed form, although illustrated, electronic or (less reliable) mental checklists exist as well.

Advantages	Explanation
Acceptance: well-known, tested, capable, effective, established and valued working aid	*Worldwide use of the method, institutional development (e.g. WHO, aviation safety culture)*
Application: efficient due to the accurate, comprehensible, uniform and brief structure; Individual or collective application (team)	*Focus on the suitability for the intended application → otherwise revision and adaptation Individual procedure or with team members*
Acronyms for specific checklists serve as mnemonic for recalling the right steps.	*Examples from aviation/medicine: IMSAFE, PAVE, TRAMP*
Teamwork: integrates, coordinates and respects all team members and their contribution by design, facilitates effective teamwork and mutual understanding	*Acknowledges the professional significance of teamwork, levels out differences, can enhance communication*
Workload: reduction of complexity and time-consumption, clear, unmistakable allocation of tasks, facilitation of professional cooperation, coordination and adequate, transparent division of labor, smooth shift handover and delegation of tasks	*Transparent, purposeful segmentation of complex tasks, workflows or processes in reasonable (sub-)steps, apportioning of tasks and responsibility. Documentation of completion. (Clarifies briefly, who does/did/will do what, why, in which order, how and when.)*
Support: help for staying focused in stressful situations, after interruptions or distractions	*Prevents negligent actions, omission or forgetting of important points/interdependencies*
Training of new staff members	*Supportive measure for introductive training of new staff, facilitates independent work*
Reference guide and orientation in complex tasks/processes/extensive timespans, relieves memory, support in non-normal situations	*Contains essential information for reference in compact form, helps keeping track of the current status, structures processes/projects/tasks*
Documentation and protection for cases of error or complaints, for internal and organizational requirements, comparison, quality control, adherence to quality and safety standards	*Legal record for possible lawsuits, verification of completion and compliance with due diligence obligations and standards, control that no prescribed steps are skipped/forgotten (safety culture/just culture)*
Cost: comparably inexpensive method	*relatively uncomplicated development*
Uncomplicated adjustment to changing requirements with revised versions, easy translation into other languages, adapted formats	*Improvement of the checklist according to demands of the organization, the users, learning from mistakes, or to factual changes → continuous reuse of an existing checklist is possible*
Criticism	
Perception as a burden of even more paper-work and an artificial procedure, application for the procedure's sake, use requires prior knowledge, dependence and reliance on checklists instead of knowledge and skills	

Table 1. Advantages of Checklist Application

The benefits of checklist implementation are obvious, but their effectiveness finally depends on the acceptance of the user. Insofar, it is on the one hand crucial to convince[2] and to train staff members and on the other hand to design checklists appropriately. Basically, there are two variants of checklists: *do-confirm* and *read-do* types, whereby the concrete development follows their intended usage.

	read-do checklist	**do-confirm checklist challenge and response** checklist
method	***read listed item → carry out task*** working step-by-step according to instruction	***carry out task → confirm completion*** working from memory, relatively free in sequence
purpose	concise summary of repetitive tasks in processes requiring the exact completion of individual steps in required sequence, guidance through the process, emphasis on precision and accuracy	verification of task completion, to avoid overlooking of important steps in complex processes, segmentation and reduction of complexity, emphasis on timely application, correctness and completeness
examples	fixed procedures, instructions, recipes	pre-flight checklists

Table 2. Main Types of Checklists

According to the organization's objectives and particular requirements, the matching type of checklist is chosen. Due to the variety of relevant tasks, a considerable number of such documents can accumulate. Therefore, it is worthwhile, to add the collection of developed checklists to an organization's process library that contains useful explanations and comments as well. Larger organizations have staff, who is responsible for checklists, their consistent development and proper application. As mentioned above, the creation of a checklist is not necessarily complicated, since templates are effortless available.
Usually, the well-known tabular layout consists of a column with checkboxes on the left-hand side of a page and next to it a column containing the list of items. Other checklists may lack the checkboxes – often in aviation related lists – because they are read and confirmed aloud. The clear arrangement of the content is very important with regard to easy legibility. Though, the formal design might seem trivial at first sight, the careful and thoughtful proceeding to create a sound checklist cannot be underestimated. The document is a synoptic summary of *only* essential items. Insofar, the creation requires a clear understanding of tasks or processes and their components.[3] Detailed instructions and explanations or comments are intentionally left out, but partially, references to further

[2] Gawande (2022) describes in detail the difficulties in overcoming reluctance and in convincing users and their organization to integrate checklists into their clinical work.

[3] Gawande (2022) recommends having five (but not more than ten) items on a checklist.

information is included.[4] If a specific sequence is required, the respective chronological order has to be considered. In order to keep the document short, it is clear that checklist application requires prior knowledge, experience and skills of the user.[5] KVWL (2024) recommends the PDCA cycle to systematically create a professional checklist.[6]

1) ***Plan:*** *identification of the problem (root causes), problem analysis, collection of relevant information and brainstorming about the process, systematization of gathered ideas and information, derivation of a conclusive written document, evaluation and control of the checklist.*
2) ***Do:*** *Approval and implementation of the developed checklist.*
3) ***Check:*** *Review of the precedent stage's effectiveness. The practical application reveals whether the checklist is suitable, effective, accepted by the intended users and if possible deficiencies (missing items, order, layout, difficulties in application, imprecisions) can be detected. Again, compilation of problems (root causes and solutions), identification of ideas for improvement and lessons learned.*
4) ***Act:*** *Improvement and adaptation of the checklist according to the results of the precedent check-phase.*
   If the result is satisfactory → end of the procedure
   If the result is not satisfactory → new PDCA cycle

The recommendations for checklist development according to Boorman (Nuclino Blog, 2019) consist of four essential aspects, whereby the first *"Investigate your failures" (1)* relates to the mentioned planning phase of the PDCA cycle. For Boorman "friction points" in routines become the starting point for a new checklist. The second advice is *"Focus on the 'stupid' stuff" (2)*, which means not going into detail, but providing precisely only the necessary, basic key points. Gawande (2022) emphasizes this criterion as well, since many avoidable (medical) errors result from rather simple causes.[7] The third aspect focuses on the tangible side of the checklist: *"Keep it simple" (3)*. Boorman summarizes, that the content shall be presented in a compact manner which does not exceed one page. Unnecessary color shall be avoided and in order to prevent distraction and to promote easy readability; a sans serif font is recommended. The fourth criterion covers the mentioned dichotomy regarding the general type: *"Decide between a 'do-confirm' or a 'read-do' checklist" (4)*. The first is rather suitable for verification after completion of the task from memory. The second type is recommended for tasks that require high precision. As seen, the significance of user-friendliness cannot be underestimated.[8] The orientation on those, who work with the checklist, has many facets which range from legibility, readability, format, material to the circumstances of application (e.g. during night-time). Checklists and their aim a closely tied to their acceptance, discipline and correct use. The following recommendations from OAS (2001) relate to the occurrence of avoidable aviation accidents caused by incorrect application or failure in checklist use: "Make a habit of using checklists consistently. Do your checklist when the workload is low. Avoid distracting conversation when performing checklists. Treat any checklist interruption as a red flag that could cause you to miss a critical item." (OAS, 2001)

Undoubtedly, these recommendations are generally valid for the application of any checklist. The last two points refer to disturbances and interruptions in performing a checklist which can lead to errors. Linde et al. (1987) investigated from a linguistic point of view interruptions during checklist application in a cockpit. According to their analysis, it is key to manage in particular the actual duration of the interruption *(hold)* and to resume *(continue)* the checklist. The authors remark that it is not completely under the control of the crew if and how often disturbances (radio communication, other tasks in the cockpit etc.) occur, but how long it takes until focus is guided again on the checklist can rather be influenced. Other, seemingly trivial, conditions can hinder the correct completion as well: missing of the checklist, using the wrong document, insufficient readability or practical inconvenience. If the application is perceived as a time-consuming burden or additional workload, the intention is taken ad absurdum, because users might tend to disregard the checklist completely.

[4] Needless to say, checklists shall be free from superfluous content and any distraction. But, the cuff checklists of the astronauts from Apollo 12 were adorned with cartoons from staff members and diverse photographs. The backup commander for the Apollo12 mission Scott explains: "We spent a lot of time going through the checklist to see where we could insert something humorous. We got that centerfold off the newsstand. Then we had to get it printed on fire-proof plastic-coated paper" (Theophanides, 2011). See Jones (1996) and Theophanides (2011) for detailed checklist photographs and comments. Additionally, Boorman, the technical lead of the development of the 787's pilot system of checklists, remarks, that "an occasional emoji is a guilty pleasure we allow ourselves to indulge in" (Nuclino Blog, 2019).

[5] Nevertheless, the KVWL publication recommends checklists for training purposes of new staff in healthcare. (KVWL, 2024, p. 96)

[6] See also Bulsuk (2009) and Skhmot (2017) about the PDCA cycle.

[7] Gawande (2022) provides examples for these simple causes of error like: lack of orderly disinfection, forgetting to wash hands, not counting materials after surgery, surgery on the wrong patient etc. Checklist application can effectively prevent such easily avoidable mistakes.

[8] The NASA published a comprehensive document that is dedicated specifically to typographic aspects concerning checklists. It covers among others topics like legibility of print (font and font size), readability (italics, bolding), color and contrast (visibility depending on illumination), opacity of the paper, lamination/glare, quality of the print and viewing abilities of the user. – Degani (1992) provides examples of fatal aviation accidents caused by difficulties with checklist application (poor readability, lack of consistency). Therefore, he stresses the relevance of the quality of documents used on the flight-deck. See Degani (1992) for more details about checklist typography.

Checklist Application in Aviation

The beginning of aviation checklists is often associated with the accident of a B-17 in 1935, where experienced and trained test pilots crashed with the bomber during climb. The accident was obviously caused by pilot error due to the complexity of the new aircraft. Technical flaws of the aircraft were not responsible for the accident. As a consequence, Boeing developed a pilot checklist prescribing concretely the duties of the pilot and copilot in order not to miss any important detail to fly the plane safely.[9] The solution proved to be – compared to the considerable damage – simple but effective; since then flights with the B-17 were safe. The obvious advantages of aviation checklists spilled over to spaceflight due to their success and became an essential component of the NASA safety culture. Astronaut Michael Collins, part of the Apollo 11 mission, even coined the expression of the "fourth crew member" which underlined the essential contribution of checklists for the mission.[10] Hersch (2009) explains, that

"[a]stronauts and engineers of the National Aeronautics and Space Administration brought manuals and checklists with them from aviation where they were already well established; in space they proliferated. Composed in a language approximating English but mostly incomprehensible to the uninitiated, in-flight documentation has been the key to the complex technologies aboard all of America's spacecraft" (Hersch, 2009).

Regardless the technical advancements since the early days of checklist implementation, their significance for aviation is undisputed. (This applies not only for professional aviation like commercial or military; also, for private pilots in general aviation, proper checklist application is an essential aspect of safety.) The general distinction between the mentioned common types of checklists exists in aviation as well. Additionally, the checklist application in normal and non-normal procedures is distinguished. The first refers to the regular course of the flight and the required standard operating procedures (SOPs), whereas the second relates to emergency or abnormal conditions[11]. These parameters determine the checklist category of choice. The following table summarizes the key points of both types according to the explanations on Skybrary (2024d, 2024e, 2024f).[12, 13]

	read-do checklist	**challenge-response checklist**
formats	printed paper or electronic versions of checklists	
application	**in non-normal procedures** (also: abnormal and emergency procedures)	**in normal procedures** (**SOP** and part of crew coordination)
	1) reaction from memory 2) checklist application (EAC) 3) further action (EAC)	normal operation of the aircraft in all phases of the flight; performance from memory according to cockpit flow pattern (specific sequence of memorized actions without checklist reference); specific critical items, cross-check → **challenge-response-checklist**
process	Immediate reactions to emergency or abnormal situations on board are carried out **from memory**.	PNF reads out the respective item and PF confirms the status/configuration (e.g. altimeter, flaps)
	Action taken is then confirmed by reference to the **"Emergency or Abnormal Checklist"** (EAC), which also contains **subsequent action**.	Electronic checklists: items may disappear/change color automatically after correct completion of the task, active annotation as "checked" partially possible
	(e.g. fire, engine failure, loss of cabin pressure, pilot incapacitation, worsening weather, fuel shortage)	
purpose	"[…] support flight crew airmanship and memory and ensure that all required actions are performed without omission and in an orderly manner." (Skybrary, 2024d) → **strict focus on safety** For reference: The "Quick Reference Handbook" (QRH) contains the relevant (normal/non-normal [EAC]) checklists.	

Table 3. read-do and challenge-response aviation checklists

The tabular overview highlighted the significance of effective, professional teamwork – good crew coordination – and strict adherence to SOPs as essential contributions to aviation safety at any time during operation. The correct application of **normal checklists** is an important SOP and represents a part of good flight crew discipline. (Skybrary, 2024 f) These checklists are used after having thoroughly completed from memory all parts of a SOP. Their purpose is the verification of proper accomplishment. This type of checklists is generally relevant for all phases of the flight, but especially for critical stages like takeoff, approach and landing. Normal checklists have to be initiated (requested/called for), performed and completed according to crew coordination SOPs. Skybrary (2024f) explains the routine as follows.

[9] See also Taylor (2020) for a picture of the original B-17 checklist from 1944 and comments.

[10] The detailed NASA "Apollo Stowage List" for the Apollo 11 mission in 1969 is an impressive example of the contribution of checklists. (NASA, 1969)

[11] **Emergency situation** means that "the safety of the aircraft or of persons on board or on the ground is endangered for any reason"; an **abnormal situation** represents conditions where "it is no longer possible to continue the flight using normal procedures but the safety of the aircraft or persons on board or on the ground is not in danger" (Skybrary, 2024e).

[12] PF = pilot flying, PNF = pilot non-flying

[13] Dismukes et al. (2010) discuss based on a qualitative study deviation from checklist application in cockpits and analyze how checklist discipline can be improved effectively. See also Degani et al. (1990) for comments about checklist misuse or rejection.

1) *<u>Initiation of normal checklists:</u> requested by the PF, read by PNF (if the PF fails to initiate, the PNF suggests it according to good CRM practice[14]), preferably during times of lower workload (prevention of time pressure/interruption) → requires sound time and workload management, respectful teamwork*

2) *<u>Conduction of normal checklists:</u> with challenge-and-response method[15], response of PF to critical items, less-critical items can be challenged/responded by PNF alone → standard rules and phraseology for normal checklists (purpose: reduction of ambiguity, improved crew communication)*

   Conduction of the checklist according to the specific rules until" (checklist name) checklist complete" marks the end of the procedure. Some normal checklists contain intended hold points where the list can be paused. (support by electronic displays of normal checklists available)

3) *<u>Management of interruptions:</u> in case of an interruption of a normal checklist the PF announces an explicit, formal "Hold (stop) checklist at (item)"and continues analogously "Resume (continue) checklist at (item)". (repetition of the last completed item before the interruption in order to prevent omission)*

As shown above, the management of **non-normal** cases differs from the regular proceeding. The applicable EAC handbook contains both, the relevant emergency and abnormal checklists, and prescribes actions which serve as initial response element. (EAC and the Operations Manual have to be congruent.)[16] In this context, Gordon et al. (2013) highlight an important aspect of non-normal situations: people might tend to do *something* instead of taking time to assess the problem first and then doing the *right thing*. Additionally, focus on operating the aircraft has to be maintained at any time. Therefore, the authors recapitulate a reasonable, practical approach: "One major airline had a fairly simple emergency checklist philosophy: recognizing that any emergency would raise the stress level as well as the potential for making a bad situation worse by rushing into a solution, the airline's policy was that the first step in any crisis was to first fly the airplane and then to assess the situation." (Gordon et al., p. 128)
Here, the view remains narrowed to the common application of normal checklists, since central aspects of day-to-day routine use are of particular interest.

As mentioned before, acronyms serve as **memory aids** for pilots to recall easily the steps of essential checklists.[17] – The considerable workload in a cockpit, the necessary constant situational awareness and the changing environment require a lot of attention by the crew. Therefore, mnemonics are useful for two particular reasons: Firstly, memory aids can relieve the memory in routine operations. Secondly, mnemonics help direct "the mind towards required actions during periods of uncertainty, or intense activity and/or emergency; i.e. preventing distraction from less critical issues" (Skybrary, 2024a). – The *IMSAFE* checklist is a method for self-assessment in order to verify whether a pilot is generally fit to fly. Partially, the additional E for emotion is included. This brief self-check facilitates the decision before flight whether it is safe to operate an aircraft or not. The *PAVE* checklist is used for a more complex pre-flight risk assessment and also determines whether the risks are acceptable and it is safe to conduct the flight. The document's items are mainly mandatory due to legal prescription.

1) ***IMSAFE** = **I**llness, **M**edication, **S**tress, **A**lcohol, **F**atigue, **E**ating (**E**motion).*
2) ***PAVE** = **P**ilot, **A**ircraft, en**V**ironment, **E**xternal Pressures.*

According to FAA (2022), the first step *P* is the connection of both checklists, whereby *IMSAFE* clarifies the safe physical and mental state of the pilot. Additionally, this step refers to the completeness of licenses and required certificates. Further, currency and proficiency reflect the skills and experience of the pilot. *A* relates to the aircraft and includes among others the pilot's familiarity with the aircraft, all required documents and equipment on board, fuel and the required capacities. *V* stands for the pilot's risk assessment concerning the weather, airport, terrain, airspace and conditions and time constraints. The last step *E* takes other external factors into consideration that could increase the risk of the flight. These factors, for example, include external expectations, avoidance of delays for passengers or emotional

[14] Checklist application is nonnegotiable and regulations require that the respective checklists have to be completed. So, it is the duty of each team member to insist on the proper use, as Gordon et al. emphasize (2013, p. 128).
[15] The application of challenge-response-concept reflects also the overall safety culture of aviation that recognizes the limited human capacities and the possibility of failure, because "[…] human factors principles dictate a challenge-and-response process between two crewmembers for conducting checklists and drills, in recognition of the susceptibility of memory to failure at critical moments" Skybrary (2024a).
[16] For more information about EAC see also Skybrary (2024e), (2024 g).
[17] For more details about pilot memory aids and the respective regulations (FAA) see also: FAA (2022), Skybrary (2024a) and Pilot Institute (2023).

pressure. – As seen, both checklists serve for risk identification and assessment prior to flight. Thereby, adherence to prescribed safety standards is the main goal. Of course, the application of mnemonics in aviation is not intended to replace the use of checklists. The mentioned acronyms rather serve as a mental hook for pilots and help to keep all items in mind.

Checklists are an integral part of the holistic concept of safety culture in aviation. As seen, the distinct components are closely related and not only the technical skills, but also the human factor is explicitly considered. Therefore, effective checklist application does not only depend on the correct completion of individual items, it is also the result of good crew cooperation, communication and reasonable workload management. The literature highlighted the aspect of checklist discipline which comprises regular training, strict adherence to SOPs and consequence in teamwork. Checklist use in performing routine tasks is efficient, and none of the simple "stupid stuff" is overlooked. In non-normal contexts, checklists help to guide attention effectively on the relevant issues and thereby facilitate the adequate situation's management. A well-known example is the successful landing by the pilots Sullenberger and Skiles on the Hudson River of flight AWE 1549. The value of checklists, besides other tools and technological support, cannot be underestimated; their capability to reduce or avoid human error and aviation accidents is proven.

Checklist Application in Healthcare

The metaphoric expression of the *golden hour*[18] or even the *golden five minutes* vividly illustrates how precious, scarce and critical time is in medical contexts. This particular valuable amount of time relates to the fact, that time is a critical factor especially in emergency and trauma care of injured people. The *golden hour* stands both symbolically for an extraordinary period of time and the assumption, that "trauma patients have better outcomes if they are provided definite care within 60 minutes of the occurrence of their injuries" (Lerner et al. 2001, p. 758).[19] "There is a golden hour between life and death. If you are critically injured you have less than 60 minutes to survive. You might not die right then; it may be three days or two weeks later – but something has happened in your body that is irreparable." (Cowley, A., UMMS, 2024) The popular, though partially questioned, concept stems from Cowley reflecting his experience in emergency and trauma care. Regardless the controversy in the literature, the necessity of efficient use of limited time and skill is undisputed. Similar to aviation, healthcare is among others characterized by a considerable workload under time pressure and high safety requirements. As seen in the previous section, the successful use of checklists is associated with enhanced productivity, efficient workload management, good teamwork and support for both routine and non-routine tasks. The present part focuses on the application of checklists in the medical field.[20] Therefore, a particular example of worldwide use – the Surgical Safety Checklist – is, besides other examples, presented.

The development of the **WHO Surgical Safety checklist** dates back with its beginnings to the year 2007.[21] A study aimed to investigate the effects of consequent checklist implementation on the numbers of surgical complications. Therefore, a 19-item surgical safety checklist was "designed to improve team communication and consistency of care" was to reduce complications and deaths associated with surgery (Haynes et al., 2009, p. 491). The following aspects represent the key points of the study. (Haynes et al., 2009, p. 492-493)

1) *"Data suggest that at least half of all surgical complications are avoidable."*
2) *"A growing body of evidence also links teamwork in surgery to improved outcomes, with high-functioning teams achieving significantly reduced rates of adverse events."*
3) *"On the basis of [WHO] guidelines we designed a 19-item checklist intended to be globally applicable and to reduce the rate of major surgical complications."*
4) *"We hypothesized that the implementation of the checklist and the associated culture changes it signified would reduce the rates of death and major complications after surgery in diverse settings."*

In the light of the precedent discussion of aviation checklists, parallels to CRM are obvious. The emphasis on good teamwork and a cultural change – like in aviation – in the healthcare sector are integral part of the approach.

[18] The term *golden hour* refers to photography and the beauty of sunlight during the first hour after sunrise and the hour before sunset.

[19] The authors question this widely accepted idea in their study and conclude: "Our search into the background of this term yielded little scientific evidence to support it" (Lerner et al., 2010, p. 760). Contrary is the opinion of Gawande (2007) in the context of triage. He describes the grave time criticality in taking care of injured soldiers and emphasizes the golden five minutes.

[20] The discussion of checklist implementation and regarding measurable effects is heterogenous in the healthcare literature. Proponents unanimously underline the advantages of the method and emphasize the ubiquitous application in aviation, whereby others recognize the wider medical context and criticize unadjusted transfer. See e.g.: Gordon et al. (2013) and Papoutsi et al. (2018).

[21] The publication of Haynes et al. (2009) presents the results of the *WHO's Safe Surgery Saves Lives program*. See also: WHO (2024a). The development and implementation of the 19-item surgical safety checklist is explained also in Gawande (2022).

The first item relates to the "acceptance" of medical error. Sullenberger argues, that "[i]n aviation, such rationalizations for avoidable human error were rejected long ago and replaced with the creation of a robust safety system that has now become the culture of the field" (Gordon et al., 2013, p. viii).

The checklist[22] comprises three major parts which segment the surgical process into the phases: *Sign in, Time out* and *Sign out*. The checklist "is used at three critical junctures in care: before anesthesia is administered, immediately before incision, and before the patient is taken out of the operating room" (Haynes et al., 2009, p. 493). The individual steps are intended to be performed as a team. Thereby, emphasis is put on precise and efficiently guided communication. The team members have to be introduced to each other, since there is clear evidence for better performance and cooperation, when the participants know each other – at least by their name and function. This is in particular relevant for teamwork in larger organizations with changing team constellations. The roles and duties are unequivocally assigned in the checklist. This relates also to the positions to be confirmed during the process. The following table recapitulates the central checklist stages according to Haynes et al. (2009).

Phase	Tasks
Sign in	***before introduction of anesthesia*** *team members (at least nurse + anesthesiologist) orally confirm the following:* – Verification of the patient's identity, surgical site and procedure + consent – Surgical site is marked/site marking not applicable. – Pulse oximeter is on the patient and functioning. – All team members are aware of patient's known allergies. – Evaluation of patient's airway and risk of aspiration + appropriate equipment and assistance are available. – Risk of blood loss: appropriate access to fluids is available.
Time out	***before skin incision*** *the entire team + any other participants involved orally confirm the following:* – Confirmation of introduction of all team members (name, role) – Confirmation of the patient's identity, surgical site and procedure – Review of the anticipated critical events including: critical/unexpected steps, operative duration, anticipation of blood loss (specific concerns of the anesthesiologist, confirmation that prophylactic antibiotics have been administered as prescribed, confirmation of sterility, equipment availability and other concerns) – Confirmation of display of the correct patient's essential imaging results
Sign out	***before the patient leaves the operating room*** *nurse reviews the following items aloud with the team:* – name of the procedure as recorded – that, if applicable, needle, sponge and instrument counts are complete – that the specimen (if any) is correctly labeled + patient's name – relevant equipment issues *The surgeon, nurse and anesthesiologist review aloud the key concerns for the recovery and care of the patient.*

Table 4. Elements of the WHO Surgical Safety Checklist

The introduction and application of the checklist requires indeed appropriate training of the staff members in order to ensure checklist discipline, correct adherence to the items and consistent documentation. Gawande (2022) recalls the hesitancy, reluctance and skepticism during the study, since the implementation of a new procedure – involving team members from all levels equally – requires some flexibility of the organization. (In addition, the remarkable influence of the status quo bias cannot be underestimated in such cases.) The results of the study showed measurable improvements regarding patient safety. The rates of any complication, the total rate of in-hospital deaths and the overall rates of surgical-site infection and unplanned reoperation dropped at all included sites after the introduction of the checklist.[23] Insofar, the authors conclude, that "[t]he reduction in the rates of death and complications suggests that the checklist program can improve the safety of surgical patients in diverse clinical and economic environments" (Haynes, 2009, p. 496). At the same time, it is recognized, that the improvements are not singularly associated with the checklist itself: rather a more complex change, that affects the professional mindset, established new routines

[22] A training video of the complete checklist application is available online, see: (NHS, 2019). For the written form see: (Haynes et al., 2009, p. 492).

[23] See Haynes et al. (2009) for detailed and contextualized results.

and workflows and the increased sensitivity for safety aspects, has taken place around the document's application. In sum, the positive convincing results led to the global application of the checklist (in adapted form).[24]

Papoutsi et al. (2018) discuss implementation and results of the "**Frailsafe"** checklist in twelve hospitals across the UK that was intended to improve specifically the safety of older patients with reliable frailty assessments. The results of the study confirmed the skepticism about transferability of checklists to the field of geriatric care. The authors recognize rather social barriers associated with hesitancy, rejection (perception as additional workload) and established professional hierarchies and boundaries. The authors explain, that "[f]ormalizing tasks and work processes in the form of a checklist placed increased emphasis on 'work-as-imagined' […] which some hospital teams found difficult to reconcile with 'work-as-done' in the messiness of everyday practice" (Papoutsi et al., 2018, p. 315). In conclusion, the authors state that "more attention to the socio-technical work" is required instead of the introduction of as technically perceived methods.

An important aspect of healthcare work is coming quickly to sound assessments and adequate priorities. A multitude of checklists exists in order to organize daily routines in healthcare.[25] This applies among others to patient communication, administering medication correctly, triage or for ensuring the availability of complete, functioning equipment. Similar to aviation, **acronyms** help also in medicine to recall checklist items quickly. The Medication checklist acronym *TRAMP* stands for: *T*ime, *R*oute, *A*mount, *M*edication and *P*atient. The checklist is intended to increase patient safety by supporting nurses in administering medicine correctly, since "[r]esearch on medical administration errors (MAEs) shows an error rate of 60%, 34 mainly in the form of wrong time, wrong rate, or wrong dose" (Nurselabs, 2015).

The author of this text had the opportunity to gather direct information regarding checklist familiarity and use from experienced medical professionals in informal conversations. All consulted persons confirmed the significant contribution of checklists in their day-to-day work. For example, in ambulances the standard stowage lists facilitate a smooth shift handover of the vehicle and make sure that all required equipment and consumable supplies are complete. Some respondents explained to know the checklist items by-heart due to the daily repetition. The author had the chance as well to read both a First Aid checklist of a nursing home and the adapted version of the WHO Safe Surgery checklist of a large German hospital.

The First Aid checklist was extensive and detailed. It contained elements of read-do checklists and comments. The amount of details and the layout over three pages did not contribute to gaining a quick overview. Additionally, the compact presentation affected the readability. The Safe Surgery checklist reflected clearly the structure and items of the original WHO template. In comparison, the document was shorter, clearer and more precise. Considering, that the latter covers the essentials of a complex surgery, the checklist was more precise, concisely and efficient compared to the First Aid document. Insofar, the immense efforts of the WHO checklist development are obvious. Nevertheless, the individual checklist is intended to be workable for the respective organization. – The First Aid document was only available in German. Considering the fact, that nursing homes employ also international staff with different levels of local language proficiency, the development of versions in different languages is recommendable; particularly in the light of effective patient safety in emergency situations.

SUMMARY

Checklist integration and their correct application tangibly reflect the attitude towards compliance with safety standards in aviation and healthcare. Correct application effectively reduces human failure – a major cause of often avoidable aviation accidents or preventable medical complications. Insofar, checklists mirror and ensure quality and safety standards for both branches and are at the same time a capable tool to meet these aspirations. Sullenberger clarifies, that "aviation safety was improved through more than checklists, as important as those are. Checklists alone cannot cure the current fragmentation of patient care or advert tragedies […]" (Gordon et al., 2013, p. viii). The precedent discussion showed the interwoven components of the holistic aviation safety culture, where checklists are one of many essential components. Insofar, the unadjusted interprofessional transfer across settings of only one isolated method is insufficient. This relates in particular to the mindset of CRM and the appreciation of team intelligence in aviation.

As seen, checklists are a comparably simple, though effectively applicable tools to even complex situations. Due to their capability to guide attention and to reduce ambiguity and complexity, checklists are indeed a suitable instrument to promote good decision-making. Undoubtedly, the preparation of decisions can benefit from the systematic, transparently reproduceable, controlled and unemotional proceeding. The degree of checklist integration differs between aviation, where their application is widely mandatory, and healthcare. For both branches the positive effects of correct use are measurable.

[24] Urbach et al. (2014) applied the method in a study with Canadian hospitals, whereby, the positive outcomes of Surgical Safety checklists could not be observed by the authors.

[25] For example, WHO (2024b) provides an equipment checklist for a triage area. See also Nurselabs (2015) for various pharmacological mnemonics. Checklists for rescue service are part of the comprehensible compendium by Jahn et al. (2022).

REFERENCES

Bulsuk, K. (2009) *Taking the first step with the PDCA (Plan-Do-Check-Act) cycle.* [Online] https://www.bulsuk.com/2009/02/taking-first-step-with-pdca.html [last accessed: October, 28th 2024].

Degani, A. et al. (1990) *Human Factors of Flight-Deck Checklists: The Normal Checklist.* Moffett Field, California: NASA, Ames Research Center. [Online] https://ntrs.nasa.gov/api/citations/19910017830/downloads/19910017830.pdf [last accessed: October, 15th 2024].

Degani, A. et al. (1992) *On the Typography of Flight-Deck Documentation.* Moffett Field, California: NASA, Ames Research Center. [Online] https://ntrs.nasa.gov/api/citations/19930010781/downloads/19930010781.pdf [last accessed: October, 15th 2024].

Dismukes et al. (2010) *Checklists and Monitoring in the Cockpit: Why Crucial Defenses Sometimes Fail.* Moffett Field, California: NASA, Ames Research Center. [Online] https://hsi.arc.nasa.gov/flightcognition/Publications/NASA-TM-2010-216396.pdf [last accessed: October, 25th 2024].

FAA/US Department of Transportation (2022) *The PAVE Checklist.* [Online] https://www.faa.gov/sites/faa.gov/files/2022-11/PAVE_0.pdf [last accessed: October, 22nd 2024].

Gawande, A. (2007) *Better. A Surgeon's Notes on Performance.* New York: Metropolitan Books.

Gawande, A. (2022) *The Checklist Manifesto. How to get things right*. New York: Metropolitan Books.

Gawande, A. (2007) *The Checklist. If something so simple can transform intensive care, what else can it do?* In: Annals of Medicine. The New Yorker. [Online] https://leaderlikeyou.com/wp-content/uploads/2024/04/New-Yorker-Power-of-the-Checklist.pdf [last accessed: October, 26th 2024].

Gordon, S. et al. (2012) *Beyond the Checklist. What else health care can learn from aviation teamwork and safety.* Ithaka: Cornell University Press.

Haynes, A. B. et al. (2009) *A Surgical Safety Checklist to Reduce Morbidity and Mortality in a Global Population.* New England Journal of Medicine 360 [5]: 491-499. [Online] https://dash.harvard.edu/bitstream/handle/1/38846186/Gawande_Surgical%20Safety%20Checklist%20to%20Reduce%20Morbidity%20vor.pdf?sequence=1&isAllowed=y [last accessed: October 16th 2024].

Hersch, M. H. (2009) *Checklist: The secret life of Apollo's 'fourth crewmember'.*
The Sociological Review, 57(1_suppl), 6-24. [Online] https://doi.org/10.1111/j.1467-954X.2009.01814.x [last accessed: October 26th 2024].

Jahn, M. et al. (2022) *Checklisten Rettungsdienst. Notfall- und Gefahrensituationen.* 2nd edn. München: Elsevier.

Jones, E. M. (1996) *Apollo 12 CDR Cuff Checklist. Apollo 12 Lunar Surface Journal.* [Online] https://www.nasa.gov/history/alsj/a12/cuff12.html [last accessed: October, 3rd 2024].

KVWL (2024) *Methoden und Instrumente. Checklisten.* [Online] https://www.kvwl.de/fileadmin/user_upload/pdf/Mitglieder/Qualitaetssicherung/Qualitaetsmanagement_KPQM/kpqm_5_30.pdf [last accessed: October, 15th 2024].

Lerner, E. B. et al. (2001) *The Golden Hour: Scientific Fact or Medical "Urban Legend"?*. Academic Emergency Medicine, 8: 758-760. [Online] https://onlinelbrary.wiley.com/doi/epdf/10.1111/j.1553-2712.2001.tb00201.x [last accessed: October, 16th 2024].

Linde, C. et al. (1987) *Checklist interruption and resumption: A linguistic study.* Moffett Field, California: NASA, Ames Research Center. [Online] https://ntrs.nasa.gov/api/citations/19880016964/downloads/19880016964.pdf [last accessed: October, 16th 2024].

NASA (1969) *Apollo 11 Stowage List, July 15, 1969.* [Online] https://www.nasa.gov/wp-content/uploads/static/history/afj/ap11fj/pdf/a11-stowage-list.pdf [last accessed: October, 1st 2024].

NHS, National Health Service UK, Gloucestershire Hospitals Foundation Trust (2019) *Video: WHO Surgical Safety Checklist Training.* [Online] https://youtu.be/IdgXe1qva5Y?si=-o6eMJoZJJuB-peOH [last accessed: October, 20th 2024].

Nuclino Blog (2019) *The simple genius of checklists, from B-17 to the Apollo missions.* [Online] https://blog.nuclino.com/the-simple-genius-of-checklists-from-b-17-to-the-apollo-missions [last accessed: October, 27th 2024].

Nurselabs (2015) *Administering Medication Checklist "TRAMP".* [Online] https://nurseslabs.com/pharmacology-nursing-mnemonics-tips/ [last accessed: October, 27th 2024].

OAS – United States Department of the Interior Office of Aircraft Services (2001) *Aviation Accident Prevention Bulletin.* [Online] https://www.doi.gov/sites/default/files/migrated/aviation/safety/upload/PB_2001-01.pdf [last accessed: October, 27th 2024].

Papoutsi, C. et al. (2018) *Improving patient safety for older people in acute a admissions: implementation of the Frailsafe checklist in 12 hospitals across the UK.* Age and Ageing 2018, 47: 311-317 [Online] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6016694/pdf/afx194.pdf [last accessed: October, 16th 2024].

Pilot Institute (2023) *PAVE Checklist explained.* [Online] https://pilotinstitute.com/pave-checklist/ [last accessed: October, 27th 2024].

Skhmot, N. (2017) *Using the PDCA cycle to support continuous improvement (Kaizen)* [Online] https://the-leanway.net/the-continuous-improvement-cycle-pdca [last accessed: October, 29th 2024].

Skybrary (2024 a) *Pilot Memory Aids* [Online] https://www.skybrary.aero/index.php/Pilot_Memory_Aids [last accessed: October, 2nd 2024].

Skybrary (2024 b) *Decision-Making (OGHFA BN)* [Online] https://www.skybrary.aero/index.php/Decision-Making_(OGHFA_BN) [last accessed: January, 26th 2024].

Skybrary (2024 c) *NASA Pre-flight Icing Checklist.* [Online] https://skybrary.aero/bookshelf/nasa-pre-flight-icing-checklist [last accessed: October, 11st 2024].

Skybrary (2024 d) *Checklists – Purpose and use.* [Online] https://skybrary.aero/articles/checklists-purpose-and-use [last accessed: October, 1st 2024].

Skybrary (2024 e) *Emergency or Abnormal Situation.* [Online] https://skybrary.aero/articles/emergency-or-abnormal-situation [last accessed: October, 1st 2024].

Skybrary (2024 f) *Normal Checklists and Crew Coordination (OGHFA BN)* [Online] https://skybrary.aero/articles/normal-checklists-and-crew-coordination-oghfa-bn [last accessed: October, 11st 2024].

Skybrary (2024 g) *Emergency and Abnormal Checklist.* [Online] https://skybrary.aero/articles/emergency-and-abnormal-checklist [last accessed: October, 11st 2024].

Taylor, S. (2020) *Check Lists.* (contains B-17F and G checklists from 1944). [Online] https://stephentaylorhistorian.com/wp-content/uploads/2020/04/b-17-pilot-checklist.pdf [last accessed: October 29th 2024].

Theophanidis, P. (2011) *"Seen any interesting hills and valley?"* [Online] https://aphelis.net/seen-any-interesting-hills-valley-playmates-on-the-moon-1969/ [last accessed: October, 16th 2024].

UMMS/University of Maryland Medical Center (2024) *History of the Shock Trauma Center. Tribute to R Adams Cowley, MD.* [Online] https://www.umms.org/ummc/health-services/shock-trauma/about/history [last accessed: October, 16th 2024].

WHO (2024a) *WHO and Surgical Safety. Surgical Safety Checklist.* [Online] https://www.who.int/teams/integrated-health-services/patient-safety/research/safe-surgery [last accessed: October, 9th 2024].

WHO (2024b) *Triage-Area – Equipment Checklist.* [Online] https://cdn.who.int/media/docs/default-source/integrated-health-services-(ihs)/csy/iitt/triage-essential-equipment-checklist.docx?sfvrsn=6e46595c_2 [last accessed: October, 20th 2024].

AUTHOR BIOGRAPHY

Stefanie Eichstedt received the degree of Magistra Artium in Russian Philology, General Linguistics and Aspects of Law from Humboldt-Universität zu Berlin, Technische Univerisität and Freie Universität in Berlin. The author received the MA degree in Aviation Management from TH Wildau.
steffie.eichstedt@gmx.de

Automatisierte Projektrenditenberechnung mithilfe von Power BI

Marcel Forster

HENRICHSEN4s

Germany
Regensburgerstraße 26
94315 Straubing
E-Mail:
marcel.forster@henrichsen4s.de

Professor Dr. Frank Herrmann

Ostbayerische Technische Hochschule Regensburg
Laboratory of Information Technology and Production Logistics (LIP)
Germany, Universitätstraße 31, 93053 Regensburg
E-Mail: frank.herrmann@oth-regensburg.de

ABSTRACT

Die Bearbeitung von Routinearbeiten ist ein wichtiger Bestandteil in jedem Unternehmen. So müssen beispielsweise Renditen für Projekte, nach deren Abschluss, berechnet werden. Häufig wird die Berechnung noch manuell durchgeführt, obwohl die Daten bereits in digitaler Form vorliegen. Dieses Projekt beschreibt wie die Projektrendite, am Beispiel der HENRICHSEN4s, automatisiert werden kann.

I. Einleitung

Die Projektrenditenberechnung ist ein wichtiger Bestandteil der Abschlussarbeiten eines Projektes. So sagen die Projektrendite viel darüber aus, wie gut ein Projekt aus finanzieller Sicht umgesetzt bzw. wie gut es von einem Projektleiter geleitet wurde.
Durch die Berechnung der Projektrendite ergeben sich für das Unternehmen verschiedene Vorteile. So haben Vorgesetzte die Möglichkeit defizitäre Projekte ausfindig zu machen und entsprechend zu handeln. Außerdem kann er die Arbeitsweisen von Projektleitern leichter miteinander vergleichen und kann somit diesem bei Bedarf mehr Hilfen zukommen lassen. Ein Weiterer Punkt ist das dadurch auch die unterschiedlichen Produkte eines Unternehmens analysiert werden können und somit Investitionsentscheidungen besser mit Daten untermauert werden können.
Das Projekt, auf dem dieses Papier beruht, beschäftigt sich mit einem für diesem Zweck erstellten Power BI Programm, mit dem die Projektrendite aus dem ERP-System automatisiert berechnet wird.

II. Analyse

Bevor ein Lösungskonzept ausgearbeitet werden kann, werden einige analytische Überlegungen durchgeführt. Die folgenden Abschnitte beschreiben die Ausgangssituation sowie die Problemstellung und Anforderungen an das anzufertigende System.

A. Ausgangssituation

Im Unternehmen wird die Berechnung der Projektrendite vor der Automatisierung von einem Azubi oder Werkstudenten per Hand durchgeführt. Dies geschieht meist nach Aufforderung eines Vorgesetzten. Dafür wird eine Excel-Datei mit VBA-Programmierung verwendet, welches sich die benötigten Daten aus dem ERP-System SITE zieht. Diese Daten müssen nach dem Übertragen überprüft werden. Deswegen werden alle Rechnungen, die für dieses Projekt infrage kommen im ERP-System überprüft, um sicherzustellen das alle Rechnungen korrekt erfasst worden sind. Ebenfalls müssen alle aufgewendeten Stunden für das Projekt überprüft werden, damit die Projektrendite korrekt berechnet wird. Nachdem die Projektrendite berechnet worden ist, werden sie im System hinterlegt und dem Vorgesetzten zur Verfügung gestellt. Der bisher verwendete Prozess ist in Abbildung 1 dargestellt.

B. Rahmenbedingungen

Die folgenden Unterkapitel geben einen Überblick über die Rahmenbedingungen des zu erstellenden Programms und die dafür benötigten Komponenten

1) ERP-System SITE:

Spezielle ERP-Software welche auf die Bedürfnisse von IT-Firmen zugeschnitten ist. Fast alle Geschäftsprozesse sind in der Lösung enthalten, vom Erstkontakt bis hin zum Controlling (s. [1]).

2) Projektmanagement-Tool WRIKE:

WRIKE ist ein Online-Tool für Projektmanagement und Zusammenarbeit (s. [2]).

3) Microsoft SQL Server Management Studio:

Integrierte Umgebung zum Verwalten beliebiger SQL-Infrastruktur (s. [3]).

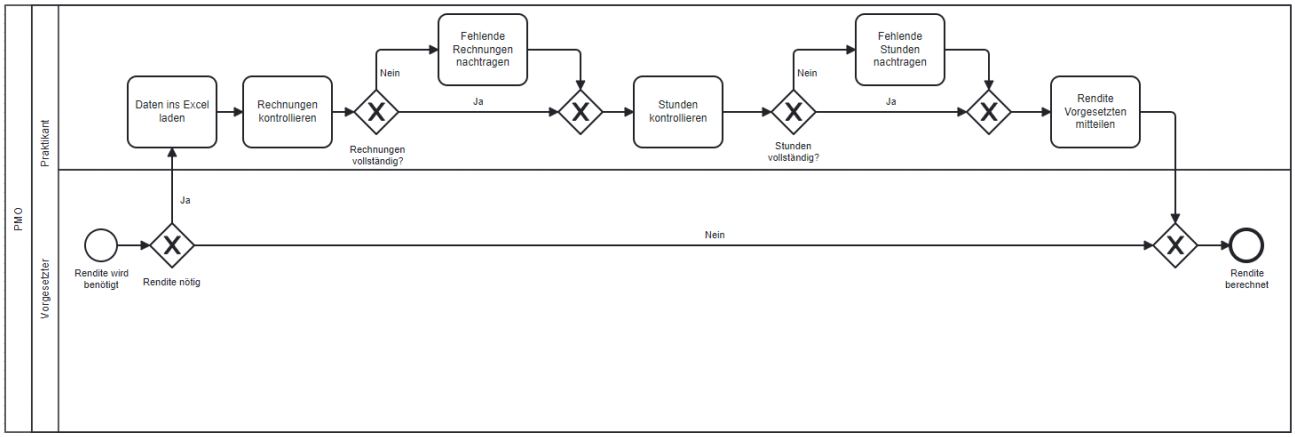


Abb. 1: Aktueller manueller Prozess der Projektrenditenberechnung

4) Power BI:

Business Itelligence-Tool von Microsoft, das zur Analyse und Visualisierung von Daten verwendet wird. (s. [4]).

C. Problemstellung

Das zu entwickelnden Programm soll die automatische Berechnung der Projektrendite für das Unternehmen ermöglichen. Hierfür sollte der vorangegangene Prozess der Projektrenditenberechnung mehrere Punkte oder Anforderungen für die Automatisierung erfüllen, um automatisiert zu werden (s. [5 & 6]).

- Manuell und wiederholbar

Damit der Prozess der Projektrenditenberechnung automatisiert werden kann, muss er zuvor manuell durchgeführt worden sein, was zuvor durch einen Azubi übernommen wurde. Wichtig ist außerdem das die Berechnung mehrmals durchgeführt wird, was mit ca. 100 benötigten Renditen pro Jahr der Fall ist.

- Regelbasiert

Um den Prozess zu automatisieren ist es nötig das er festen Regeln folgt. Dafür müssen Regeln implementiert werden damit die Berechnung immer nach den gleichen Regeln ablaufen. So muss ein Großprojekt wie ein Großprojekt berechnet werden und ein Kleinprojekt wie ein Kleinprojekt.

- Digitaler Input

Die vorhandenen Daten, welche für die regelbasierte Berechnung benötigt wird, muss digital vorliegen und auch digital wieder ausgelesen werden können. Wichtig hierbei ist das die Daten immer an der gleichen Stelle & Format gespeichert wird. Ansonsten müssen für diese Fälle Aufwändige Ausnahmen programmiert werden.

- Standardisierter Prozess

Der zu automatisierende Prozess sollte in der nahen Zukunft nicht grundlegende geändert werden oder einer kontinuierlichen Verbesserung unterliegen. Deswegen bietet es sich an den Prozess vor der Automatisierung kritisch zu begutachten und eventuelle Änderung in der Implementierungsphase miteinzufließen. Aus diesem Grund wird auch die Formel für die Projektrenditen auf ihre Richtigkeit überprüft und Änderung können gleich eingepflegt werden.

- Volumen/ROI

Damit sich die Automatisierung von einem Prozess lohnt, müssen die notwendigen Ausführungen der Projektrenditenberechnung eine bestimmte Anzahl an berechneten Renditen geben. Hierbei werden durchschnittlich 100 Projekte pro Jahr berechnet, was eine geringe Anzahl an automatischen Prozessen bedeutet. Bei einer so geringen Stückzahl ist es fraglich, ob der Return on Investment (ROI), bei einer durchschnittlichen Bearbeitungszeit pro Projekt von 45 Minuten, in einer angemessenen Zeitspanne positiv wird. Deswegen darf man den ROI nicht als einzige Kennzahl einer Automatisierung sehen, sondern muss weitere Problemstellungen mit einbeziehen wie nachfolgende.

- Fehlervermeidung

Ein weiteres Problem sind menschliche Fehler in der Prozessbearbeitung, so können oftmals unbeabsichtigt Fehler bei der Bearbeitung eines Prozesses passieren. Die sich auf weitere Entscheidungen negativ auswirken können. So kann es sein das aufgrund eines schlechten Tages oder falsch erlernten Tätigkeiten vorkommt das die Renditen einer bestimmten Projektart falsch berechnet werden und deswegen falsche Schlussfolgerungen gezogen werden. Aus diesem Grund sollte der Prozess, welcher automatisiert wird, zwar Fehleranfällig sein, allerdings sollten die Fehler, im

Idealfall, auf menschliches Versagen zurückzuführen sein und nicht auf den Prozess selbst.

- Qualität erhöhen/konstant halten

Wie im Punkt der Fehlermöglichkeit bereits erwähnt, passieren menschliche Fehler, was zu einer schlechteren Datenqualität führen kann. Durch eine Automatisierung des Prozesses lassen sich beispielsweise menschliche Rechenfehler in der Projektrendite vermeiden und die Gesamtqualität wird gesteigert. Des Weiteren werden die Qualitätspunkte eines Prozesses dauerhaft auf einem konstanten Niveau gehalten und sind nicht mehr von der täglichen Form der ausführenden Person abhängig.

- Fehlende Analysemöglichkeiten

Durch das Fehlen einer Automatisierung für einen Prozesses kommt es aufgrund der mangelnden oder unterschiedlichen Datenqualität zu keiner Analysemöglichkeiten oder nur zu einer stark eingeschränkten Möglichkeit, was wiederum für weitere Geschäftsentscheidungen schlecht sein kann. Durch die Automatisierung hat man insgesamt eine bessere Datenqualität, wodurch weitere Analysen oder Prognosen auf einem viel zuverlässigeren Fundament stehen.

- Datenschutzprobleme

Ein weiteres Problem ist der Datenschutz, oder auch Datenminimierung. So muss der Mitarbeiter, der deinen Prozess durchführt, auch die nötigen Befugnisse haben, um die benötigten Daten für die Durchführung des Prozesses einzusehen, obwohl er vielleicht nur die Befugnisse für die berechnete Rendite hat. Außerdem können bei einem manuellen Prozess die Ergebnisse wieder in einer Datei gespeichert werden, auf die weitere Mitarbeiter Zugriff haben, obwohl es sich um Ergebnisse für ausgewählte Personen handelt. Durch die Automatisierung des Prozesses müssen die benötigten Daten nur von einem Entwickler angesehen werden und können für andere gesperrt werden.

D. Anforderungen

Nachfolgend werden die Anforderungen für das zu entwickelnde Programm dargestellt:

- Nachvollziehbarkeit der Projektrendite muss durch genauere Auswahl der aufgewendeten Stunden eines Projektes gewährleistet werden
- Die Bearbeitungszeit, der Zeitraum ab wann ein Projekt abgeschlossen wurde bis zu seiner Berechnung der Rendite, muss auf kürzeste Zeit gesenkt werden.
- Die Bearbeitungszeit, die Zeit, die für eine Berechnung nötig ist, muss auf unter eine Minute gesenkt werden.
- Die manuelle Nacharbeit, bei der alle Rechnungen und erfasste Stunden kontrolliert werden, muss komplett entfallen.
- Alle manuellen Schritte wie die Kontrollen der Ergebnisse müssen automatisiert werden, damit freiwerdende Zeit besser genutzt werden kann.
- Visuelle Darstellung der Ergebnisse muss mittels graphischer Elemente erfolgen, damit Prognosen bessergestellt werden können.
- Es dürfen nur Themenprojekte, größere Projekte mit einem Projektleiter, automatisiert werden, weil hier die Logik für alle Projekte gleich sind.
- Es dürfen nur Projekte angestoßen werden auf denen bereits Zeiten geleistet worden sind.
- Es dürfen Projekte nur von einem bestimmten Mandanten (H4s-Mandant) / Zeitpunkt berechnet werden.
- Die verwendeten Daten müssen aus dem ERP-System SITE kommen und es dürfen nur benötigten Daten aus dem ERP-System entnommen werden, um die Performance zu verbessern.
- Der Datenschutz muss gewährleistet werden. Was bedeutet das nur die betreffenden Personen Zugriff auf die Power BI Auswertungen erhalten dürfen.
- Ein Weiterer Punkt in Bezug auf den Datenschutz ist das zwischen Vorgesetztem/Teamleiter und Arbeiter/Projektleiter unterschieden werden muss. Hierbei dürfen Teamleiter alle notwendigen Informationen eines Projektes sehen. Ein Projektleiter darf nur die Informationen für seine Projekt einsehen.

E. Alternativen

Neben dem im Rahmen dieses Projektes zu realisierender Software, wurde sich auch über alternative Lösungen Gedanken gemacht.
So Stand im Raum die bereits verwendete Excel Datei auf den neuesten Stand zu bringen und einen neuen Prozess einzuführen, der die Bearbeitung für die Renditeberechnung wesentlich beschleunigt. Aufgrund der Anforderungen für eine bessere Datenqualität und vor allem der visuellen Darstellung wird sich für die Lösung mittels Power Bi entschieden.

III. Lösungskonzept

In diesem Abschnitt wird sich auf die programmiertechnischen und graphischen Konzepte für die im vorherigen Kapitel genannten Anforderungen festgelegt.

A. Entwicklungsumgebung und Programmiersprache

Da für die Umsetzung Power BI verwendet wird, wird auch die Programmiersprache DAX für Power BI Code verwendet. Für Tests der vorausgewählten Daten wird Microsoft SQL verwendet und somit die Programmiersprache SQL.

B. Authentifizierung und Rollenverteilung

Um den Datenschutz umzusetzen wird ein Authentifizierungssystem verwendet. Um einen Benutzer zu identifizieren, wird das allgemeine Microsoft Konto verwendet. Um den Zugriff auf die Projektrenditen im Allgemeinen zu regeln, wird mit Zugriffsrechten gearbeitet. Um die Datenminimierung nach der DSGVO zu gewährleisten, wird eine Rollenverteilung implementiert, um zwischen Teamleiter und Projektleitern zu unterscheiden.

C. Automatische Aktualisierung der Daten

Um stets aktuelle Ergebnisse zu gewährleisten, müssen die für den Prozess benötigten Daten immer auf den aktuellen Stand sein. Aus diesem Grund muss eine beständige Aktualisierung der Daten konfiguriert werden.

D. Festlegen der durchzuführenden Tätigkeit

Um den Prozess der automatisierten Projektrenditenberechnung zu automatisieren, muss sich im Vorfeld überlegt werden welche Arbeiten zu automatisieren sind. Diese Punkte sind zum einen aus der manuellen Tätigkeit herauszuarbeiten, diese sind das Laden der benötigten Daten und die Berechnung der Renditen. Zum anderen werden die neuen Tätigkeiten aus den gestellten Anforderungen ermittelt, was der besseren Visualisierung der Ergebnisse entspricht.

- Auslesen der Daten

Die zuvor ausgeführte manuelle Tätigkeit das Auslesen der Daten muss für eine komplette Automatisierung ebenfalls automatisch erfolgen. Hierbei ist zu beachten das die Daten, welche ausgelesen werden im benötigten Format für die durchzuführenden Tätigkeiten ist. Falls das nicht der Fall ist, müssen die Daten vorher angepasst werden. Hierbei ist ebenfalls festzulegen aus welchem System die benötigten Daten für die Berechnungen kommen. Außerdem muss bestimmt werden welche Daten für die auszuführenden Tätigkeiten benötigt werden.

- Berechnen der Renditen

Wie bereits zuvor wird auch die Berechnungen für die Rendite für eine vollständige Automatisierung wieder automatisch ausgeführt. Dieses Mal aber nicht in einer Excel Datei, sondern im Power BI Programm. Hierbei werden vor allem die Berechnungen für Renditen festgelegt. Diese sind folgende:

(1) Rohertrag = Ausgangsrechnung – Ausgangsgutschriften – (Eingangsrechnungen – Eingangsgutschriften) – Kosten eigene Software
(2) Deckungsbeitrag (berechnet) = Rohertrag – (erfasste Zeiten + Weiterbildungen + MA-Pate +zusätzliche Zeiten)
(3) Deckungsbeitrag (bereinigt) = Deckungsbeitrag (berechnet) – (erfasste Wegzeiten + angeordnete Kulanz + Garantie + Servicezeiten + Vertriebsunterstützungszeiten)
(4) Projektrendite (berechnet) = [Deckungsbeitrag (berechnet)] / [Ausgansrechnungen – Ausgangsgutschriften - Eingangsgutschriften]
(5) Projektrendite (bereinigt) = [Deckungsbeitrag (bereinigt)] / [Ausgansrechnungen – Ausgangsgutschriften - Eingangsgutschriften]

- Visualisieren der Renditen

Wäre für eine mögliche visuelle Darstellung der Daten früher eine zusätzliche Datei nötig in der die benötigten Diagramme erstellt worden wären, sollen sie mit dem neuen Programm direkt in Power BI mit den berechneten Ergebnissen automatisch nach einer Vorlage erstellt werden. So wird festgelegt das die Daten auf insgesamt 2 Hauptseiten und 2 Detailseiten angezeigt werden. Die dafür benötigten Diagramme werden im vorherein festgelegt, um dann mit den berechneten Daten befüllt zu werden

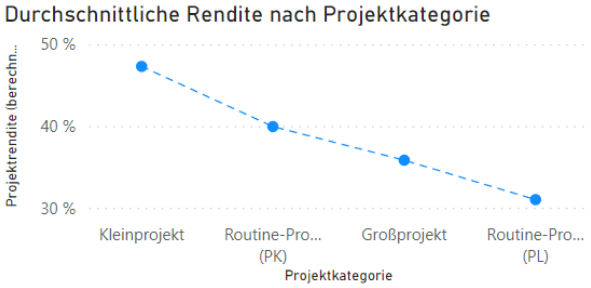


Abb. 2: Beispiel des Diagrammes für die durchschnittliche Rendite nach der Projektkategorie

E. Festlegung der Daten

Bevor ein Prozess automatisiert werden kann, müssen zuerst die dafür benötigten Daten für die auszuführenden Daten festgelegt werden. Diese unterscheiden sich je nach Anwendungsfall erheblich. Deswegen müssen für die automatisierte Projektrenditenberechnung die Daten ausgewählt werden, welche für die zu berechnenden Renditen notwendig sind. Aus diesem Grund werden die benötigten Daten nach der Formel für Projektrendite herausgesucht. Diese sind im ERP-System SITTE:

- Projekte
- Dienstleistungsaufträge
- Eingangsrechnungen
- Eingangsgutschriften
- Ausgangsrechnungen
- Ausgangsgutschriften
- Verkaufschancen
- Verkaufsaufträge
- Herstellercode

F. Überprüfen der Datenqualität

Im nachfolgenden Unterkapitel wird die Datenqualität anhand 6 verschiedener Kriterien beurteilt. Die Datenqualität ist von entscheidender Bedeutung für eine Automatisierung eines Prozesses, in diesem Fall der Projektrenditenberechnung. Nur durch eine entsprechende Datenqualität lässt sich eine Automatisierung durchführen, weil nur dann ein Standardisierter Prozess abgearbeitet werden kann und immer die gleichen Ergebnisse auf Grundlage der gleichen Daten berechnet werden.

- Relevanz

Bei der Relevanz der Daten wird überprüft, inwieweit der Informationsgehalt mit dem Informationsbedarf des Anwenders übereinstimmt. So ist eine Bereitstellung der Daten zu 100% für den Anwender nicht nötig, wenn es sich dabei um Daten handelt, welche er nicht benötigt. So benötigt man beispielsweise keinen Status eines Projektes, wenn für die Berechnung nur nummerische Daten nötig sind. Hier ist es so, dass im ERP-System viele Daten gespeichert sind, die für die Berechnung nicht benötigt werden.

- Konsistenz

Bei der Konsistenz der Daten wird geprüft, ob man die Daten Widerspruchsfrei aus unterschiedlichen Anwendungen bekommt. Dies ist für eine Automatisierung sehr wichtig, weil sonst die Daten falsch ausgelesen werden und somit auch die Ergebnisse falsch wären und es somit zu einem Glaubwürdigkeitsproblem kommen würde. Deswegen müssen alle Daten auf ihre Konsistenz hin überprüft werden. Damit eine Rechnung eindeutig einem Projekt zugeordnet werden kann. Allerdings sind beispielsweise nicht allen Rechnungen immer einer Projektnummer eindeutig zugeordnet.

- Zuverlässigkeit

Die Zuverlässigkeit beschreibt die ausreichende Verfügbarkeit von Informationen, um die ursprüngliche Herkunft und die verschiedenen Transformationen der Daten nachvollziehen zu können. Die Zuverlässigkeit der Daten kann gewährleistet werden, weil die Daten nur im ERP-System geändert werden können.

- Korrektheit

Bei der Korrektheit der Daten muss geprüft werden, ob die Daten die inhaltlich korrekte Information in den benötigten Formaten beinhaltet. Denn wenn bei einer Automatisierung die Formatierung in den benötigten Feldern nicht gegeben ist, werden dadurch falsche Ergebnisse berechnet, weil das Zahlenformat nicht übereinstimmt. So muss in diesem Beispiel dafür gesorgt werden das Zeiten, welche in Tage angegeben sind, immer zuerst in Stunden umgerechnet werden, damit bei der Berechnung kein Einheitenfehler unterläuft.

- Vollständigkeit

Die Vollständigkeit der Daten beschreibt ob für alle benötigten Daten auch Daten hinterlegt sind und keine NULL-Werte oder leere Datenfelder. Für eine Automatisierung ist es wichtig, dass es keine leeren Felder gibt, weil ansonsten Berechnungen nicht oder falsch ausgeführt werden. Allerdings verhält es sich ähnlich wie die bei der Konsistenz der Daten das die Daten bei den Rechnungen nicht immer eingehalten wird.

- Zeitnähe

Bei der Zeitnähe muss zum einen der Anlieferungszeitpunkt der Daten festgelegt werden und zum anderen die Aktualität der Daten. Die Aktualität der Daten ist durch die Eingabe der Nutzer in das ERP-System gegeben und die Abfragehäufigkeit kann in Power BI bis hin zu einer Echtzeitdarstellung der Daten eingestellt werden.

Nach der Überprüfung ist aufgefallen das nur die Vollständigkeit der Daten im ERP-System gewährleistet werden kann. Außerdem kann die Zeitnähe der Daten durch einfache Einstellungen in Power BI gewährleistet werden. Die restlichen vier Punkten können nicht gewährleistet werden, was wiederum auf eine schlechte Datenqualität zurückzuführen ist. Dadurch kann auch keine Automatisierung gewährleistet werden, weil ansonsten fehlerhafte Berechnungen aufgrund von fehlerhaften Daten zustande kommen würden. Aus diesem Grund muss die Datenqualität verbessert werden, bevor die Daten in Power BI geladen und die Berechnung automatisiert werden. Da die Daten in einer Datenbank

hinterlegt sind bietet es sich an die Daten mittels SQL aufzubereiten.

IV. Implementierung

Wie in der Datenqualität erwähnt, müssen die Daten für eine automatische Projektrenditenberechnung zuerst mittels SQL-Befehlen aufbereitet werden. Um möglichst wenige Daten für die Automatisierung zu verwenden, um damit die Perfomance zu verbessern, werden nur die benötigten Daten nach Power BI geladen. Dafür werden die benötigten Daten ebenfalls mit SQL ausgewählt.

A. SQL-Befehle

1) Projekt

Um die notwendigen Informationen für das jeweilige Projekt zu bekommen, wird aus dem ERP-System die hinterlegten Daten für das Projekt herausgelesen. Diese sind die Projektnummer, der Projektleiter und der Prozesscode. Um nur die Informationen gemäß den Anforderungen auszulesen, werden nur Themenprojekte und Daten vom neuen H4s-Mandanten ausgelesen.

```
SELECT
    job.[No_] AS Projektnummer,
    job.[Person Responsible] AS Projektleiter,
    job.[Process Code] AS Prozesscode

FROM [SITE2021-Prod].[dbo].[H4s_Produktiv$Job$437dbf0e-84ff-417a-9
    WHERE job.[Person Responsible] <> '' AND
    job.[Process Code] LIKE 'PJ-THEMEN' AND job.[No_] LIKE 'SPR%'
```

Abb. 3: Beispiel für SQL-Statement für die Projekte

2) Dienstleistungsaufträge

Um die benötigten Daten für die erfassten Zeiten zu erhalten, werden die Dienstleistungsaufträge ausgelesen. Hierbei sind die benötigten Daten auf 5 verschiedene Tabellen in der Datenbank verteilt, welche mittels Joins zusammengefügt werden. Da allerdings nicht alle Daten von Dienstleistungsaufträge benötigt werden, weil es sich beispielsweise um Daten von nicht Themenprojekte handelt, müssen diese Daten weiter gefiltert werden. So werden nur die DLA-Nummer, die Re.-Nr., die Beschreibungen, das Leistungsdatum, der Arbeitstyp, die Menge, der Berechnungsart, der Projektnummer und der Bestellnummer. Zusätzlich werden die Daten für die Konsistenz vereinheitlich damit für die spätere Aktualisierung keine Einheitenfehler vorhanden sind. Um nur die Daten für den aktuellen Mandanten zu bekommen, wird mittels Where-Abfragen sichergestellt, das auch nur die benötigten Daten ausgewählt werden.

3) Eingangsrechnungen

Bei der Eingangsrechnung, welche die Leistungen von externem Dienstleister an uns berechnet, werden die Daten für die eigene Rechnungsnummer, externen Rechnungsnummer, Kreditor, Rechnungsdatum, Zeilenbetrag, Einheitencode, Sachkonten, Projektnummer benötigt und mittels Select-Befehls aus drei verschiedenen Tabellen ausgelesen. Mittels Where-Abfragen werden nur Rechnungen, die für die entsprechenden Themenprojekte benötigt werden, ausgelesen und somit die Relevanz der Daten sichergestellt.

4) Eingangsgutschriften

Falls es Gutschriften für eine externe Rechnung gibt, werden für diese ebenfalls die Gutschriftennummer, die externe Gutschriftennummer, der Kreditor, das Sachkonto, das Rechnungsdatum, die Menge der Zeilenbetrag, die Beschreibung und die Projektnummer ausgelesen werden. Ebenfalls dürfen aufgrund von Datensparsamkeit nur die benötigten Daten für die bestimmten Projekte ausgewählt werden.

5) Ausgangsrechnungen

Für die Ausgangsrechnungen, in welchem die erbrachte Leistung an den Kunden berechnet werden, müssen nachfolgende Daten ausgelesen werden, wie die Rechnungsnummer, der Zeilenbetrag, die Ressourcennummer, der Herstellercode, der Einheitencode, der Einstandspreis, der Projektnummer, und das Datum. Außerdem müssen die Zeiten, welche abgerechnet werden in ein einheitliches Format gebracht werden, weil die Abrechnung auf Basis des Angebotes erfolgt, welches oft Pauschalen oder Tage als Zeiteinheiten verwendet, aber die geleisteten Zeiten werden als Stunden erfasst. Ebenso dürfen nur Daten für die benötigten Projekte ausgelesen werden.

6) Ausgangsgutschriften

Ebenfalls wie bei Eingangsgutschriften gibt es auch Gutschriften, wenn fehlerhafte Rechnungen erstellt worden sind. Hier müssen die Informationen wie die Gutschriftennummer, die Projektnummer, die Menge, welche wieder auf ein einheitliches Format gebracht werden, der Zeilenbetrag, die Ressourcennummer und das Buchungsdatum ausgelesen. Allerdings nur für die benötigten Projekte.

7) Verkaufschancen

Verkaufschancen sind die ersten Angebote, welche ein Verkäufer an einen Kunden macht, und diese bestimmen im maßgeblichen Sinne die Rendite, je nachdem wie viel Rabatt gewährleistet wird. Für die weitere Berechnung müssen alle Daten wie Einkaufspreis und Verkaufspreis der verkauften Dienstleistung ausgelesen werden. Da die Kunden in den meisten Fällen verhandeln gibt es mehrere Versionen einer Verkaufschance, weswegen die erste Version ausgelesen werden, welche das ursprüngliche Angebot enthält.

8) Verkaufsaufträge

Nachdem mit dem Kunden eine Einigung über die zu erbringende Leistung und dem Preis erzielt worden ist, wird diese in einem Verkaufsauftrag festgehalten. Deswegen müssen für die Berechnung die Daten wie der Auftragsnummer, Zeilenbetrag, Beschreibung, Menge, Einheitencode, Projektnummer, Einstandspreis ausgelesen werden. Ebenfalls aber nur die Daten, welche für Themenprojekte bestimmt sind.

9) Herstellercode

Da in fast jedem Projekt Software verkauft wird, müssen bei den Rechnungen die Hersteller mit angegeben werden. Da sich die Rendite bei selbstentwickelter Software anders verhält wie bei gekaufter Software müssen sie deswegen ausgelesen werden. In der Datenbank werden sie in anderen Tabellen, nicht bei den Rechnungen, gespeichert. Deswegen müssen die Rechnungsnummer, der Zeilenbetrag und der Herstellercode ausgelesen werden.

Nachdem alle SQL-Statements erstellt sind, um nur die benötigten Daten in der erforderlichen Qualität und Konsistenz auszulesen, werden diese mittels DirectQuery in Power BI eingepflegt. So ergeben sich insgesamt über 10 Tabellen, welche die benötigten Informationen enthalten. Durch die SQL-Befehl werden teilweise bis zu 80% der benötigten Datenmenge eingespart, weil nur Daten für Themenprojekte geladen werden.

B. Beziehungen von Tabellen in Power BI

Damit für spätere Darstellungen und Auswertungen, die Daten auch entsprechend interagieren können müssen sie mittels Beziehungen verbunden werden. Hierbei ist wichtig das die Daten in der richtigen Richtung miteinander verbunden sind. So muss die Kardinalität festgelegt werden, wobei es in diesem Fall nur zwei Möglichkeiten gibt (1:1, M:N). Diese legt fest welche Richtung die Beziehung hat. So wird die Projektnummer von einem Projekt mit den Projektnummer der erfassten Zeiten verbunden. Mit den erfassten Zeiten werden alle anderen Tabellen verbunden, die mithilfe der vorherigen SQL-Statements erzeugt wurden. Hierbei ist zu beachten das die Beziehung universalgültig sind, sich also auf die übergeordnete Instanz bezieht und nicht auf einen bestimmten Wert. Dies ist für die automatisierte Anwendung wichtig, weil mit der Zeit immer mehr neue Projektnummern oder andere nummerische Werte erzeugt werden, die das System dann automatisch zuordnet.

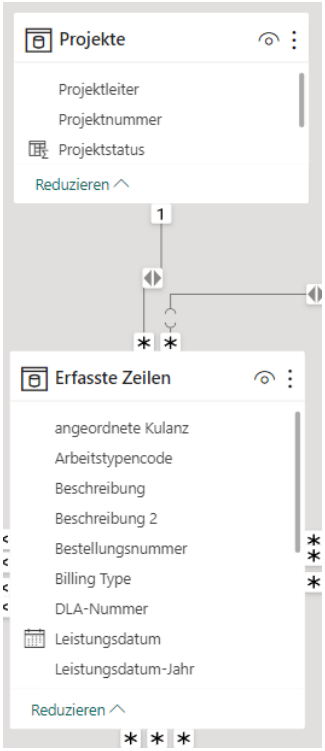


Abb. 4: Beispiel einer Beziehung von Projekt und Erfassten Zeiten

Nachdem alle benötigten Tabellen vorhanden sind, müssen noch weitere Tabellen erstellt werden. So sind viele Daten aus der Datenbank nur in numerischen Werten angegeben. Für eine bessere Lesbarkeit und spätere Bearbeitung werden Mapping-Tabellen erstellt. Diese sollen beispielsweise den Arbeitstyp, welcher in der Datenbank mit null bis 8 angegeben ist, in eine leserliche Form bringen.

C. Berechnungen in Power BI

Nachdem alle Tabellen in Power BI mit der richtigen Verknüpfung verbunden sind, müssen die Formeln für die Berechnung eingefügt werden. Auch hier ist wieder wichtig das die Berechnung allgemeingültig ist, dass auch zukünftige Werte korrekt berechnet werden. Für die korrekte Berechnung ist es aber noch notwendig alle verschiedenen Daten richtig zu identifizieren, welche sich oftmals aus einer oder zwei Spalten zusammensetzt.

1) Berechnung einzelner Zwischenergebnisse

- Summe erfasste Zeiten:
Umfasst alle Zeiten, die für ein Projekt aufgewendet wurden.
- H4s interne Zeiten:
Umfasst alle Zeiten, die von der HENRICHSEN4s geleistet wurden.
- Externe Zeiten:
Umfasst alle Zeiten, die von externen Dienstleistern geleistet wurden.
- Zugeordneter Aufwand:
Umfasst alle Zeiten, die dem Kunden berechnet werden können.
- Kulanz:
Umfasst alle Zeiten, die der Projektleiter auf Kulanz genommen hat und nicht an den Kunden berechnet werden können.
- Garantie:

Umfasst alle Zeiten, welche aufgrund von Garantieansprüchen geleistet werden müssen.

- Intern:

Umfasst alle Zeiten welche intern für das Projekt geleitstet wurden aber nicht verrechnet werden.

- Angeordnete Kulanz:

Umfasst alle Zeiten, welche zugeordnet wurden und vom Teamleiter angeordnet wurden.

- Zusätzlich:

Umfasst alle Zeiten, welche nicht in vollem Umfang geleistet wurden, aber abgerechnet werden können.

- Weg:

Umfasst alle Zeiten, die von den Mitarbeitern für Wege zum Kunden angefallen sind.

- Vertriebsunterstützung:

Umfasst alle Zeiten, die vom Vertrieb geleistet wurden.

- Servicezeiten:

Umfasst alle Zeiten, die vom Service Team geleistet wurden.

- Weiterbildungen:

Umfasst alle internen Zeiten, welche für Weiterbildungen verwendet wurden.

- MA-Pate:

Umfasst alle internen Zeiten, welche für Mitarbeiter Patenschaft verwendet wurden.

Diese herausgearbeiteten Daten werden in einer eigenen virtuellen Tabelle in Power BI gespeichert, wo sie für weitere Berechnungen zur Verfügung stehen. Wobei man für die Automatisierung herausstellen kann, dass alle benötigten Daten durch bestimmte Merkmale eindeutig identifiziert werden müssen.

2) Berechnung der Projektrendite

Um die Projektrendite mit den Formeln aus Kapitel III D) zu berechnen werden diese nun mittels Measures in Power BI übertragen. Hierbei werden folgende Werte berechnet:

- Rohertrag:

Gibt die Kosten der Handelspanne eines Unternehmens an

- Deckungsbeitrag (berechnet):

Gibt die Differenz zwischen den dem Rohertrag und variablen Kosten an

- Deckungsbeitrag (bereinigt):

Gibt die Differenz zwischen den dem Rohertrag und variablen Kosten an, die ein Projektleiter nicht beeinflussen kann

- Projektrendite (berechnet):

Gibt die Rendite für das Projekt für das Unternehmen an

- Projektrendite (bereinigt):

Gibt die Rendite für das Projekt für den Projektleiter an unter Berücksichtigung bestimmter Parameter

- Score:

Kennzahl wie gut ein Projekt von einem Projektleiter unter Berücksichtigung verschiedenere Umstände geleitet wurde

```
Projektrendite (berechnet) = 'Berechnung'[Deckungsbeitrag (berechnet)]
/(SUM('Ausgangsrechnungen'[Zeilenbetrag ohne MwSt.])
- SUM('Ausgangsgutschriften'[Zeilenbetrag ohne MwSt.])
- SUM(Eingangsgutschriften[Zeilenbetrag ohne MwSt.]))
```

Abb. 5: Beispiel einer Formel in Power BI

D. Visualisierung

Um die Ergebnisse besser als bisher zu visualisieren, werden die Diagramme ebenfalls in Power BI erstellt. Hierbei werden vier verschieden Seiten eingerichtet. Auf der ersten Seite werden alle Renditen aller Projekte angezeigt. Auf der zweiten Seite werden die Scores angezeigt. Hierbei werden die Renditen und Scores, aus den vorhandenen Daten berechnet. Die erstellten Diagramme werden jeweils bei der Rendite und den Scores nach der Projektart, Projektkategorie und Projektleiter dargestellt. Die Grundlage der Diagramme sind die berechneten Werte, dadurch werden die Diagramme automatisch angepasst, wenn sich die zugrundeliegenden Daten ändern.
Mittels Drilltroughs werden die Kennzahlenseite, in der alle Zeiten entsprechend der vorherigen Berechnung aufgeführt sind, und die Detailseite, in der alle Details aufgeführt sind, für ein einzelnes Projekt generiert.

F. Berechtigungssystem & Rollenmodell

Um bei den Auswertungen und Berechnungen auch den Datenschutz zu berücksichtigen, wird ein Berechtigungssystem und ein Rollenmodell implementiert.

1. Berechtigungssystem

Damit nicht mehr wie früher die berechneten Renditen an einem zentralen Ort abgespeichert werden, wo jeder Zugriff hat, wird ein Berechtigungssystem eingeführt. Dieses System regelt wer grundsätzlich auf die automatisiert berechneten Ergebnisse zugreifen kann. So werden allen Teamleitern und Projektleiter Leserechte gewährt, um auf die entsprechenden Seiten zuzugreifen. Die Authentifizierung und Zuteilung der Accounts der betroffenen Personen funktioniert über die eingebaute Authentifizierung von Microsoft. So ist der Zugriff direkt mit dem Benutzerprofil in der Firma verbunden und es müssen keine weiteren Authentifizierungen durchgeführt werden.

2. Rollenmodell

Zweitens wird ein Rollenmodell implementiert, welches festlegt was die zugriffsberechtigen Personen jeweils sehen dürfen. Aufgrund der DSGVO Vorschriften dürfen eben nur berechtigte Personen bestimmte Daten einsehen. Um das zu gewährleisten und den Personen nur die benötigten Daten zur Verfügung zu stellen wird die Projektrendite an den Projektleiter gekoppelt. Deswegen wird für jede zugriffsberechtigte Personen eine Rolle definiert bzw. angelegt. So gibt es die Rolle des Admins

für alle Teamleiter, weil diese aufgrund ihrer Arbeit alle Ergebnisse sehen müssen. Für alle anderen werden eigene Rollen eingerichtet. Für diese Rollen wird die Microsoft Authentifizierung mit dem Projektleiter aus SITE verbunden. Dadurch sehen die Projektleiter nur die Projekte, bei denen sie in SITE hinterlegt sind.

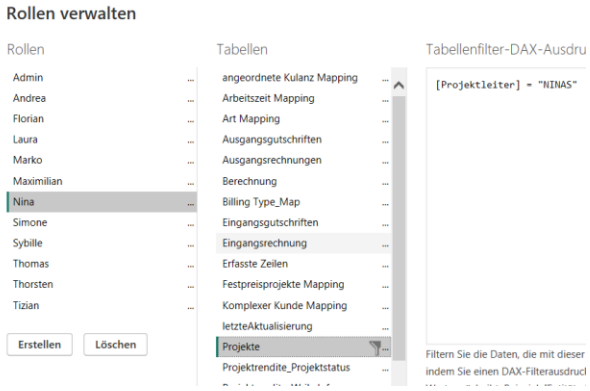


Abb. 6: Zuteilung einer Rolle an einem Projektleiter

G. Aktualisierungsintervall

Um die zugrundeliegenden Daten für die Berechnungen auf dem aktuellen Stand zu halten, wird eine automatische Aktualisierung benötigt. Dazu muss erst eine dauerhafte Verbindung zu der zugrundeliegenden Datenbank erstellt werden. Hierfür wird eine Gateway-Verbindung auf die Datenbank eingerichtet.

Abb. 7: Gateway Verbindung mit Datenbank

Daneben gibt es verschiedene Möglichkeiten, um die Aktualität der Daten zu gewährleisten. So können bereits bei den SQL-Befehlen zwischen DirectQuerys, direkter Zugriff auf die Daten, oder Import, benötigte Daten werden in bestimmten Intervallen als Kopie aus der Datenbank gezogen, ausgewählt werden. Für den Anwendungsfall der Projektrenditenberechnung wurde die Option des Imports aufgrund der Datenmenge und der Relevanz der erzielten Ergebnisse gewählt, weil keine sekundengenaue Berechnung nötig ist. Stattdessen wurde ein tägliches Aktualisierungsintervall eingeführt, bei dem alle Daten einmal täglich vor Arbeitsbeginn in die Anwendung geladen werden und alle Berechnungen ausgeführt werden.

V. Ergebnisse und erzielte Verbesserungen

Durch die automatische Auswahl der Daten und automatische Ausführung der Tätigkeiten, wie die Berechnung und die Erstellung von Diagrammen wird die manuelle Arbeit vollständig ersetzt und somit werden Arbeitszeiten frei, die auf andere Aufgaben gelenkt werden können. So werden durchschnittlich ca. 45 Minuten pro Projektrenditenberechnung eingespart, was bei ca. 100 jährlich zu berechnenden Renditen eine Zeitersparnis von 75 Stunden bedeutet. Außerdem wird die Rendite nun einmal täglich berechnet sodass bei einem abgeschlossenen Projekt die Rendite spätestens nach 24 Stunden zur Verfügung steht, was neue Möglichkeiten für weitergehende Analysen entstehen lässt. So kann zukünftig direkt nach Projektende überprüft werden ob die geplanten Zeiten auch mit den tatsächlichen Zeiten übereinstimmt oder es Abweichungen gibt. Als letzten Punkt lässt sich die genauere Verteilung der Stunden nennen, durch die man genauer erkennen kann, warum eine Rendite überdurchschnittlich gut oder schlecht ist. Aus diesem genauen Datensatz lässt sich zukünftig ableiten, ob es Auffälligkeiten bei einem Projektleiter gibt, und man kann ihn punktgenau unterstützen. Oder man kann ableiten das es bei bestimmten Kunden immer wieder zu bestimmten Mehraufwände kommt und kann dann für zukünftige Projekte entsprechende Maßnahmen einplanen. Insgesamt konnte die Datenqualität stark gesteigert werden und dadurch auch die Analysefähigkeit der Führungsebene. Durch die Implementierung der Berechtigungssystems kann es zukünftig zu keinem versehentlichem Datenschutzverletzung kommen und man muss nicht mehr befürchten, dass deswegen ein Prozess überarbeitet werden muss.

VI. Zusammenfassung

Zusammenfassend lässt sich sagen, dass eine Automatisierung viele Vorteile mit sich bringt. Um aber eine Automatisierung umzusetzen, um daraus Vorteile zu generieren, müssen im Vorfeld viele Punkte beachtet werden. So muss als erster die Ausganssituation genau untersucht werden, ob eine Automatisierung sinnvoll ist oder nicht. Dafür werden folgende Punkte begutachtet: Als erstes wird überprüft, ob die benötigten Daten bereits digital hinterlegt sind. Zweitens wird überprüft ob durch eine Automatisierung Einsparungen in Form von Zeit, Geld oder andere Zugewinne bedeuten. Des Weiteren muss geklärt werden, ob es sich bei dem Prozess der Automatisiert werden soll um eine standardisierte und wiederholbare Aufgabe handelt. Können diese Fragen positiv beantwortet werden, muss ein Konzept erstellt werden, in dem alle benötigten Tätigkeiten aufgeführt werden, welche automatisiert werden sollen. In diesem Fall war, dass das Laden der Daten und die Berechnung der Projektrendite. Als nächsten Punkt müssen alle benötigten Daten durch eindeutige Kennzeichen maschinell identifiziert werden können. Darüber hinaus ist eine solche Automatisierung eine gute Möglichkeit, um den Prozess auf seine Aktualität zu überprüfen oder Änderungen einzupflegen. In diesem Fall wurden die Formeln auf ihre Aktualität überprüft und angepasst. Ebenso wurden neue Diagramme, welche automatisch aus den Daten erzeugt werden, eingebaut. Außerdem konnte die Datenqualität insgesamt gesteigert werden,

weil es menschliches Versagen nur noch in der Implementierungsphase gibt und ansonsten genau nach dem Prozess vorgegangen wird. Um diese Fehler zu minimieren ist eine Testphase erforderlich. Als letzten Punkt für Automatisierung kann noch die Einhaltung des Datenschutzes genannt werden. So konnte in diesem Beispiel eine Berechtigungssystem eingebaut werden sodass nur noch berechtigte Personen auf die Ergebnisse Zugriff haben.

VII. References

[1] Singhammer, "SITE - die ERP-Software speziell für IT-Firmen", https://www.singhammer.com/erp-software-site/, Abgerufen am 15.09.2023

[2] WRIKE, "Hindernisse beseitigen, Klarheit schaffen, Ziele übertreffen", https://www.wrike.com/de/, Abgerufen am 15.09.2023

[3] Microsoft, "Was ist SQL Server Management Studio (SSMS)", https://learn.microsoft.com/de-de/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16, Abgerufen am 15.09.2023

[4] Microsoft, "Microsoft Power BI: Datenvisualisierung", https://powerbi.microsoft.com/de-de/, Abgerufen am 15.09.2023

[5] K. Lizenberger, "5 RPA Voraussetzungen", https://nativdigital.com/rpa-voraussetzungen/ Abgerufen am 15.09.2023

[6] LINXYS, "Die vielen Vorteile der Automatisierung von Geschäftsprozessen", https://www.linxys.ch/die-vielen-vorteile-der-automatisierung-von-geschaeftsprozessen/, Abgerufen am 15.09.2023

Möglichkeiten und Grenzen KI-unterstützter Softwareentwicklung im SAP-Umfeld

Konstantin Schatz
T.CON GmbH & Co. KG
OTH Regensburg
Straubingerstraße 2
94447 Plattling
konstantin.schatz@st.oth-regensburg.de

Norbert Kölbl
Solution Architect
T.CON GmbH & Co. KG
Straubingerstraße 2
94447 Plattling
norbert.koelbl@team-con.de

Prof. Dr. Frank Herrmann
Ostbayerische Technische Hochschule Regensburg
Labor für Informationstechnik und Produktionslogistik (LIP)
Galgenbergstraße 32
93053 Regensburg
frank.herrmann@oth-regensburg.de

ABSTRACT

Generative Künstliche Intelligenz wird zunehmend auch in der Softwareentwicklung eingesetzt. Verschiedene Anbieter, wie GitHub, eine renommierte Entwicklerplattform, haben Lösungen herausgebracht, die durch Codevervollständigung den Softwareentwicklungsprozess beschleunigen und fehlerfrei machen sollen. Doch der Einsatz dieser Lösungen birgt Risiken. Wem gehört der generierte Code? Was geschieht mit den Daten, anhand derer der Prompt Code generiert?
Ziel ist es, Bedürfnisse an Codegenerierungstools für die Firma T. CON zu definieren, und die drei Lösungen GitHub Copilot, Tabnine und Codeium anhand der Bedürfnisse miteinander zu vergleichen.

Keywords

Analyse, ABAP, Eclipse, Entwicklung, Generative KI, JavaScript, KI, LLM, ML, Python, SAP, Software, UI5, VS Code

1. EINLEITUNG

1.1 Einleitung

Laut Thomas Dohmke, CEO der weltweit größten Entwicklungsplattform GitHub, wird bereits (Stand Januar 2024) die Hälfte aller Software mit Unterstützung von KI-Copiloten entwickelt, mit der Prognose, dass dieser Anteil in den nächsten Jahren auf 80% ansteigt. Laut eigener Aussage können Entwickler einen Produktivitätsgewinn von 30 bis 50 Prozent verzeichnen. [1] Auch die SAP kündigte bei der TechEd 2023 an, jeden Entwickler, zum Entwickler für generative KI zu machen. Die Roadmap ist klar: Ohne den Einsatz generativer KI soll es bei der SAP keine Softwareentwicklung mehr geben.[2] Künstliche Intelligenz findet immer weiter Einzug, sowohl in unser Privatleben als auch in Unternehmen. Nach Prognose der International Data Corporation verfünffachen sich die Ausgaben für KI-Plattformen zwischen 2022 und 2026. [3] Deutsche Unternehmen betrachten laut Bitkom e. V. künstliche Intelligenz zu 65% als Chance, jedoch benutzen sie nur 18% des Mittelstands. Jedoch sind viele Erwartungen an KI und generative KI laut Gartner zur Zeit überzogen. [4]

Als technologieoffenes Unternehmen möchte die T. CON Möglichkeiten und Grenzen neuer Technologien erforschen, um sie sicher zu benutzen.[5]
Allerdings möchte die T. CON nicht ausschließlich abhängig von SAP-eigenen Tools sein. Daher soll in diesem Projekt erörtert werden, welche SAP-fernen generative KI-Tools sich für den Einsatz in der Softwareentwicklung eignen.

1.1 Problemstellung und Zielsetzung

Das Hauptziel und Kernergebnis des Projektes besteht darin, mit Hilfe einer wissenschaftlichen Evaluation festzustellen, ob der GitHub Copilot, Tabnine und Codeium sich für die Bedürfnisse der T. CON eignen.
Daher ist ein Unterziel mit Hilfe von Experteninterviews herauszufinden, welche Bedürfnisse die Softwareentwickler bei der T. CON haben. Dies beinhaltet unter anderem die Benutzung von IDEs und Programmiersprachen, sowie urheberrechtliche Aspekte. Auf den Ergebnissen der Experteninterviews aufbauend sollen dann in einer Nutzwertanalyse die drei Tools GitHub Copilot, Tabnine und Codeium analysiert werden. Dafür müssen aussagekräftige Vergleichskriterien gefunden werden. Die Evaluierung der Codequalität der Tools wird mit Hilfe von prototypischen Implementierungen durchgeführt.

2. T. CON

Die T. CON ist ein 1999 gegründetes SAP-Beratungsunternehmen mit Hauptsitz in Plattling und weiteren Standorten in Regensburg, Passau, Heilbronn, Hamburg und Berlin. Die 420 Mitarbeiter betreuen 240 Kunden durch Kompletteinführungen, Optimierungen, S/4-Transitionen, Managed Services und Digitalisierung, was ihnen bereits eine Auszeichnung für exzellente MILL-Expertise durch die SAP eingebracht hat. Es gilt als Familienfreundlich und ist regelmäßig unter den „Bayerns BEST 50". Das agile Unternehmen strebt neues Wachstum an, wirtschaftlich wie personell. [6]

3. PROBLEMANALYSE

Um die geeigneten Tools zu analysieren, ob sie sich für die T. CON eignen, müssen zunächst die Bedürfnisse der Firma erörtert werden. Hierfür wird die Methodik des Experteninterviews verwendet. Diese ist eine qualitative Forschungsmethode, bei der eine oder mehrere Personen interviewt werden, welche über spezifisches Fachwissen verfügen. [7] In diesem Projekt eignet sich das Experten-interview, da die Bedürfnisse der T. CON spezialisiertes Wissen sind, welches nicht in Literatur zu finden ist.
In der Problemstellung ist bereits aufgeführt, dass die drei Tools GitHub Copilot, Tabnine und Codeium analysiert werden. Dabei war der GitHub Copilot bereits von der Firma vorgegeben. Die anderen beiden Tools ergaben sich auf folgenden Gründen: Um zwischen mehreren Tools auswählen zu können reicht die Bewertung dieses Tools allein nicht. Jedoch ist auch die Bewertung von mehr als drei Tools aufgrund des zeitlichen Rahmens nicht sinnvoll, weswegen sich auf drei Tools geeinigt wurde.
Mindestanforderungen an ein Tool war, dass es direkt in VS Code integriert werden kann, das ergab sich aus den Experteninterviews. Die Entscheidung fiel in enger Absprache mit dem Firmenbetreuer auf zusätzlich Tabnine und Codeium, Konkurrenzprodukte des GitHub Copilot. Sobald die Bedürfnisse klar sind, müssen die Tools analysiert werden. Hierfür eignet sich die Nutzwertanalyse. Mit dieser Methodik können objektive Bewertungen hinsichtlich mehrerer unabhängiger Kriterien vorgenommen werden. Die Nutzwertanalyse ist eine strukturierte Methodik, die große Vorteile zur Entscheidungsfindung bietet. Die klar festgelegte Vorgehensweise gewährleistet eine bessere Nachvollziehbarkeit des Entscheidungsprozesses. Auch ihre quantitative Ausrichtung macht sie objektiver. Jedoch hängen die Genauigkeit der Ergebnisse von den verfügbaren Daten ab. Außerdem bietet die Gewichtung der Kriterien die Gefahr der Subjektivität. [8]

4. GRUNDLAGEN UND BEGRIFFSERKLÄRUNGEN

Bevor die Lösung erarbeitet wird, werden wichtige Begriffe definiert für das weitere Verständnis in der Abschlussarbeit.

4.1 SAPUI5

SAPUI5 ist ein von SAP entwickeltes Toolkit für die Webentwicklung, basierend auf JavaScript, HTML5 und CSS3. Es ermöglicht die einfache Erstellung von Fiori-Apps und dient als Grundlage für verschiedene Benutzeroberflächen in SAP-Lösungen. Entwickler nutzen Tools und Bibliotheken von SAPUI5 für die responsive Entwicklung von Webanwendungen, die sowohl auf Desktops als auch auf mobilen Geräten lauffähig sind [9]. Es ist der Nachfolger von SAP-UI-Entwicklungstechnologien wie BSP, PDK, Web Dynpro Java und Web Dynpro ABAP. Die Open-Source-Variante OpenUI5 ermöglicht auch Nicht-SAP-Kunden die Entwicklung plattformunabhängiger Anwendungen. Insgesamt erleichtert SAPUI5 die effiziente und benutzerfreundliche Webentwicklung für verschiedene Geräte und Plattformen [10].

4.2 Künstliche Intelligenz

Künstliche Intelligenz (KI) ist ein Teilgebiet der Informatik, das sich auf die Entwicklung von Maschinen konzentriert, die intelligentes Verhalten zeigen. KI umfasst alle Anstrengungen, deren Ziel es ist, Maschinen intelligent zu machen. Dabei wird Intelligenz verstanden als die Eigenschaft, die ein Wesen befähigt, angemessen und vorausschauend in seiner Umgebung zu agieren. Dies beinhaltet die Fähigkeit, Sinneseindrücke wahrzunehmen und darauf zu reagieren, Informationen aufzunehmen, zu verarbeiten und als Wissen zu speichern, Sprache zu verstehen und zu erzeugen, Probleme zu lösen und Ziele zu erreichen.[11]

Machine Learning
Machine Learning (ML) ist ein Teilbereich der Künstlichen Intelligenz (KI), der sich auf die Entwicklung von Algorithmen konzentriert, die es einem Computer ermöglichen, aus Erfahrungen (Daten) zu lernen und sich zu verbessern, ohne explizit programmiert zu sein. ML-Algorithmen bauen ein statistisches Modell auf, das auf Trainingsdaten beruht und welches gegen die Testdaten getestet wird. Es werden nicht einfach die Beispiele auswendig gelernt, sondern Muster und Gesetzmäßigkeiten in den Lerndaten erkannt.
Es gibt verschiedene Arten von maschinellem Lernen, darunter überwachtes Lernen, unüberwachtes Lernen, halbüberwachtes Lernen und bestärkendes Lernen. Beim überwachten Lernen wird das Modell mit Eingabe-Ausgabe-Paaren trainiert und das Ziel ist es, eine Funktion zu lernen, die Eingaben auf Ausgaben abbildet. Beim unüberwachten Lernen hingegen stehen keine Ausgabedaten zur Verfügung und das Ziel ist es, die Struktur in den Eingabedaten zu finden. Halbüberwachtes Lernen ist eine Kombination aus überwachtem und unüberwachtem Lernen, bei dem sowohl markierte als auch unmarkierte Daten für das Training verwendet werden. Beim bestärkenden Lernen lernt ein Agent, wie er sich in einer Umgebung verhalten soll, um eine Belohnung zu maximieren. [12]

Natural Language Processing
Natural Language Processing (NLP) bezieht sich auf den Zweig der Informatik – und insbesondere der KI – der sich damit befasst, Computern die Fähigkeit zu geben, Text und gesprochene Sprache auf dieselbe Art und Weise zu verstehen wie Menschen. NLP kombiniert Computerlinguistik — regelbasiertes Modellieren natürlicher Sprache — mit statistischem maschinellem Lernen und Deep-Learning-Modellen. Es umfasst eine Reihe von Techniken zur Interpretation menschlicher Sprache, oft mit dem Ziel, menschliche Sprache so zu verstehen und zu manipulieren, wie es Menschen tun. NLP umfasst sowohl die Verarbeitung (Verstehen) als auch die Generierung (Erzeugung) von natürlicher Sprache.[13]

Large Language Model
Large Language Models (LLMs) sind eine Art von maschinellem Lernen Modell, das darauf abzielt, menschenähnliche Texte zu erzeugen. Sie sind eine Unterklasse von Natural Language Processing (NLP) Modellen und verwenden Techniken aus dem Bereich des Deep Learning, um große Mengen an Textdaten zu verarbeiten und zu lernen, wie man menschenähnliche Texte erzeugt.
LLMs sind in der Regel neuronale Netzwerke, die auf einer Architektur namens Transformer basieren. Transformer-Modelle verwenden eine Technik namens "Aufmerksamkeit", um zu bestimmen, welche Teile eines Textes für die Vorhersage des nächsten Wortes relevant sind. Dies ermöglicht es ihnen, den Kontext über lange Distanzen hinweg zu berücksichtigen und komplexere Muster in den Daten zu erkennen. [14]

4.3 Datenschutz

SOC 2
Die SOC 2-Zertifizierung, entwickelt vom American Institute of Certified Public Accountants (AICPA), konzentriert sich auf die Sicherheit, Verfügbarkeit, Verarbeitungsintegrität, Vertraulichkeit und Datenschutz von Informationen in Serviceorganisationen. [15] Es definiert fünf Trust Service Criteria: Sicherheit, Verfügbarkeit, Integrität der Verarbeitung, Vertraulichkeit und Datenschutz. Diese Kriterien sind grundlegend für die Sicherheit und Integrität von Daten. SOC 2-Konformität dient nicht nur als Compliance-Label, sondern auch als Vertrauensanker für Kunden und als Wettbewerbsvorteil für Unternehmen, insbesondere in den Bereichen IT-Dienstleistungen und Cloud-Services. Es trägt auch wesentlich zum Risiko-management bei, indem es Unternehmen hilft, Sicherheitslücken zu identifizieren und zu schließen, das Risiko von Datenschutzverletzungen zu minimieren und gesetzliche Anforderungen zu erfüllen. Die SOC 2-Zertifizierung wird durch Certified Public Accountants in einem sogenannten SOC 2-Audit durchgeführt. [16]

DSGVO
Die Datenschutz-Grundverordnung (DSGVO) ist eine rechtliche Rahmenrichtlinie der Europäischen Union, die am 25. Mai 2018 in Kraft trat. Ihr Hauptziel ist der Schutz personenbezogener Daten innerhalb der EU und die Festlegung einheitlicher Standards für deren Verarbeitung. Die DSGVO ersetzt die vorherige Datenschutzrichtlinie von 1995 und berücksichtigt technologische Entwicklungen. Ein zentraler Aspekt ist die extraterritoriale Anwendbarkeit auf Organisationen, die Daten von EU-Bürgern verarbeiten, unabhängig vom Standort. Die Verordnung definiert klar personen-bezogene Daten und gewährt betroffenen Personen umfassende Rechte über ihre Daten. Rechtmäßigkeit, Fairness und Transparenz sind grundlegende Prinzipien, und die Verordnung betont Datenminimierung, Integrität und Vertraulichkeit. Verantwortliche und Auftrags-verarbeiter müssen strenge Anforderungen erfüllen, und die DSGVO sieht Geldbußen für Verstöße vor. Die Einführung der DSGVO markiert einen Paradigmenwechsel im Datenschutzrecht, indem sie den Schutz personenbezogener Daten als grundlegendes Menschenrecht anerkennt und Organisationen zu proaktiven Sicherheitsmaßnahmen verpflichtet. [17]

4.4 Herausforderung bei KI-unterstützter Softwareentwicklung

Vorgabe im Zuge der Projektarbeit war es auch, sich kritisch mit der Benutzung von generativen KI-Tools auseinanderzusetzen. Drei große Punkte sind ins-besondere bei der Verwendung von Codevervoll-ständigungstools zu beachten. Halluzinationen, die zur Erzeugung syntaktisch unkorrekten Codes führen kann, KI-Demenz, die das Sprachmodell zukünftiger Generationen verschlechtern kann, und urheberrechtliche Aspekte.

Halluzination
In der Künstlichen Intelligenz (KI) bezieht sich das Phänomen der Halluzination auf die Tendenz von Modellen, insbesondere von Large Language Models (LLMs), Informationen zu erzeugen, die nicht in den Eingabedaten vorhanden sind. Diese "halluzinierten" Informationen können in Form von zusätzlichen Details, falschen Behauptungen oder inkorrekten Interpretationen auftreten. [18]
Halluzinationen treten häufig auf, wenn ein Modell versucht, eine Aufgabe zu erfüllen, für die es nicht ausreichend trainiert wurde oder die über seine Fähigkeiten hinausgeht. Zum Beispiel könnte ein Modell, das darauf trainiert wurde, Texte zu generieren, "halluzinieren", indem es Details hinzufügt, die nicht in den ursprünglichen Eingabedaten vorhanden waren, oder indem es Behauptungen aufstellt, die nicht durch die Daten gestützt werden.
Das Phänomen der Halluzination ist ein aktives Forschungsgebiet in der KI. Forscher versuchen, die Ursachen für Halluzinationen zu verstehen und Methoden zu entwickeln, um sie zu verhindern oder zu minimieren. Einige Ansätze beinhalten die Verbesserung der Trainingsdaten, die Verwendung von Techniken zur Überwachung und Kontrolle der Modellausgabe und die Entwicklung von Methoden zur Erkennung und Korrektur von Halluzinationen. [19]
Halluzinationen können ein Zeichen dafür sein, dass ein Modell über seine Grenzen hinaus arbeitet. Sie können auch dazu führen, dass ein Modell ungenaue, irre-führende oder sogar schädliche Informationen erzeugt. Daher ist es wichtig, bei der Verwendung von KI-Modellen Vorsicht walten zu lassen und sicherzustellen, dass die Modelle ordnungsgemäß überwacht und kontrolliert werden. [20]

KI-Demenz
Ein weiterer bedeutender Aspekt bei der Anwendung künstlicher Intelligenz ist das Phänomen der KI-Demenz, das in KI-Modellen auftritt, die auf künstlich generiertem Text basieren. Eine wissenschaftliche Untersuchung von Stanford University und University of California, Berkeley, zeigte, dass GPT-4 im März 2023 bestimmte Aufgaben effektiver bewältigte als im Juni desselben Jahres. Beispielsweise erkannte GPT-4 im März 2023 97,6% aller Primzahlen richtig, doch dieser Anteil sank bis Juni 2023 auf 2,4%. Formatierungsfehler bei der Generierung von Programmcode nahmen ebenfalls zu, was darauf hinweist, dass das Verhalten eines Sprachmodells in kurzer Zeit erheblichen Veränderungen unterliegen kann. [21]
Die Qualität künstlicher Intelligenz hängt wesentlich von der Güte des zugrunde liegenden Datensatzes ab. Von Menschen erstellte Datensätze gewinnen an Bedeutung, wie Forscher von Oxford, Cambridge und London feststellten. Trotz vielversprechender Ergebnisse durch Training mit von Menschen generierten Daten besteht die Prognose, dass zukünftige Sprachmodelle auch mit Daten trainiert werden, die von vorherigen Modellen generiert wurden. Dies könnte zu irreversiblen Defekten führen, wie in einer Studie beobachtet, wo am Ende kein sinnvoller Text mehr generiert wurde.[22] Professor Arvind Narayanan von der Princeton University kritisiert die Studie, da die Verschlechterung der Ergebnisse möglicherweise auf die gestellten Aufgaben, insbesondere das Schreiben von Programmcodes und das Lösen von mathematischen Aufgaben, zurückzuführen sei. OpenAI selbst bestreitet einen Leistungsabfall und argumentiert, dass Anwender die Qualität der Sprachmodelle subjektiv und im Laufe der Zeit kritischer bewerten. Die mangelnde Offenlegung von Trainings- und Modifikationsdetails durch Open-AI erschwert jedoch Vorhersagen zur zukünftigen Entwicklung. [23]
Im Kontext der KI-unterstützten Softwareentwicklung bleibt zu beobachten, wie sich verschiedene Sprachmodelle entwickeln. Von der Verschlechterung des Modells GPT-4 wäre zunächst GitHub Copilot betroffen, Tabnine, Codeium oder SAP-Modelle könnten zukünftig auch betroffen sein. Insgesamt erfordert das Phänomen der KI-Demenz, ähnlich wie Halluzination, eine sorgfältige Überwachung der Modelle.

Urheberrechtsaspekte
Urheberrechtsaspekte sind in der KI-unterstützten Softwareentwicklung relevant in zwei Richtungen: Die generierten Inhalte können urheberrechtlich geschützt sein, jedoch können Programme, die hauptsächlich durch KI entwickelt wurden, möglicherweise keinen Schutz mehr genießen. [24]
Die Verwendung urheberrechtlich geschützter Trainingsdaten für KI-Modelle kann zu rechtlichen Problemen führen. Ein aktuelles Beispiel ist eine Klage der New York Times gegen OpenAI wegen der Verwendung urheberrechtlich geschützter Zeitungsartikel für das Training von Sprachmodellen. [25]
Es ist unklar, wie diese Fälle ausgehen werden und welche Auswirkungen sie auf den deutschen Rechtsraum haben werden. Darüber hinaus sind die generierten Inhalte von KI nicht automatisch urheberrechtlich geschützt, was zu weiteren rechtlichen Komplikationen führen kann, insbesondere wenn sie mit eigenen kreativen Werken vermischt werden. [26]

4.5 Analysierte Tools

GitHub Copilot, ein KI-basiertes Code-Completion- und Code-Generierungs-Tool, wurde in Kooperation zwischen GitHub und OpenAI entwickelt.[27] Es fungiert als Erweiterung für Visual Studio Code und analysiert öffentlich verfügbaren Code auf GitHub, um Entwicklern kontextbezogene Vorschläge für die Codevervollständigung zu bieten.[28] Basierend auf GPT-3 ermöglicht GitHub Copilot nicht nur einfache Code-Snippets, sondern auch komplexe Funktionen und Algorithmen. Es wird von über 27 Prozent aller Entwickler, insbesondere von Python-Entwicklern (40 Prozent), genutzt, wobei mehr als ein Viertel der Codevorschläge angenommen wird. [29]
Tabnine ist ein weiterer KI-basierter Codevervollständigungsassistent, der die allgemeine Produktivität steigern soll. Mit Funktionen wie automatischer Vervollständigung, Kommentieren und Datenschutz kann Tabnine in verschiedenen Programmiersprachen und IDEs eingesetzt werden.[30] Die KI-basierte Codevervollständigung kann lokal ausgeführt werden, ohne persönliche Daten oder Codes zu teilen. Tabnine respektiert laut eigenen Angaben den Datenschutz und Sicherheit und verwendet keine nicht erlaubten Quellcodes für sein Training.[31]
Codeium ist ein KI-basiertes Tool, das als Programmierassistent fungiert und intelligente Vorschläge für Codegenerierung und Navigation bietet. Es unterstützt über 70 Programmiersprachen und kann in bevorzugten Code-Editoren installiert werden. Mit Funktionen wie Autocomplete und Chat optimiert Codeium den Codierungsprozess. Es respektiert nach eigenen Angaben ebenfalls Privatsphäre und Sicherheit, speichert oder verkauft keine persönlichen Daten oder Codes und verwendet keine nicht erlaubten Quellcodes für sein Training. Codeium bietet kostenlose und kostenpflichtige Optionen für individuelle und Teamnutzung. [32]

5. EXPERTENINTERVIEWS

Mit den Experteninterviews wurden in diesem Projekt zwei konkrete Ziele verfolgt:
Das erste Ziel war von der Firma vorgegeben: Die Softwareentwickler bei der T. CON sollen für die neuen Technologien an denen erforscht werden sensibilisiert werden, und es soll erörtert werden, wie der Wissenstand über generative KI im Unternehmen ist. Dieses Wissen ist allerdings nicht relevant für die spätere Nutzwertanalyse und wird daher nicht näher erläutert.
Das zweite Ziel war es, konkrete Anforderungen an ein Tool für die spätere Nutzwertanalyse herauszufinden. Das sind zunächst die IDEs, welche verwendet werden. Außerdem wird erörtert, mit welchen Programmiersprachen in welchen Umgebungen programmiert wird.

5.1 Interviewleitfaden

Der Interviewleitfaden ist mit dem Betreuer der Abschlussarbeit erstellt worden. Er enthält die zu stellenden Fragen, und mögliche Rückfragen. Er erleichtert später beim Interview auf verschiedene Antworten unterschiedlich zu reagieren, um dem Interview einen größeren Nutzen zu bieten. [7] Die ersten drei Fragen dienen dazu, Kenntnisse der Mitarbeiter bezüglich der Thematik der Abschlussarbeit zu erlangen. Anschließend wird nach den benutzten Programmiersprachen und Entwicklungsumgebungen gefragt. Danach sollen die Experten ihre Meinung zu Vor- und Nachteilen KI-unterstützter Softwareentwicklung nennen, sowie, welche Datenschutz- und Complianceanforderungen wichtig sind. Schließlich noch die bereits genannte Möglichkeit, Wünsche und Anregungen zu nennen. Es ist üblich, den Interviewleitfaden einem Pre-Test zu unterziehen, um sicherzustellen, dass Fragen verständlich formuliert sind. In regelmäßigen Abstimmungen mit dem Firmenbetreuer wurden die Fragen auf Verständlichkeit geprüft, jedoch wurde auf umfangreiche Pre-Tests verzichtet.

5.2 Kontaktierung der Interviewpartner

Die Auswahl der Interviewpartner hat Einfluss auf die Güte der Datenerhebung. [7] Sie erfolgte in Abstimmung mit dem Betreuer. Die interviewten Personen stellen einen Querschnitt der Softwareentwickler bei der T. CON dar. Sie sind im September 2023 über Microsoft Teams interviewt worden. Die 20-minütigen Interviews beinhalteten eine Präsentation des GitHub Copiloten, um einen ersten Eindruck der Experten zu bekommen.

5.3 Durchführung

Die Durchführung beinhaltete einen Dank für die Teilnahme, eine Einverständniserklärung zur Aufnahme des Interviews, sowie eine kurze Beschreibung des Ziels des Interviews. Am Schluss wird noch auf einen auszufüllenden Fragebogen hingewiesen.

5.4 Transkription und Kodierung

Die Aufzeichnungen wurden wörtlich transkribiert, wobei auf Füllwörter wie "Äh" verzichtet wurde. Das Transkript wurde nicht gekürzt, jedoch wurden Namen und andere Angaben anonymisiert, um die Vertraulichkeitsvereinbarung zu gewährleisten. Anschließend wird die unübersichtliche und umfangreiche Materialsammlung codiert. Dabei werden die Kernaussagen identifiziert. Die Codierung erfolgte pro Antwort in Tabellenform.

5.5 Ergebnis der Interviews

Die Ergebnisse der Experteninterviews beinhalten nicht nur die Kernaussagen, sondern bestimmen das weitere Vorgehen der Abschlussarbeit.

Programmiersprachen
Acht Experten bei der T. CON arbeiten mit ABAP, sieben mit JavaScript, vier mit TypeScript, drei mit Python, zwei mit HTML/XML, und je ein Experte mit C++ und C. JavaScript und TypeScript sind ausschließlich im UI5-Umfeld anzufinden.
Für die Integration von KI-Tools in die Softwareentwicklung liegt der Fokus auf JavaScript, TypeScript und Python. ABAP wurde in Absprache mit dem Firmenbetreuer bei der anschließenden Bewertung der KI-Tools außen vorgelassen, da die Hoffnung auf SAP interne Tools in der Zukunft besteht, und in einem anderen Projekt bereits ein firmeninternes generatives KI-Modell auf ABAP-Code gebaut wird.

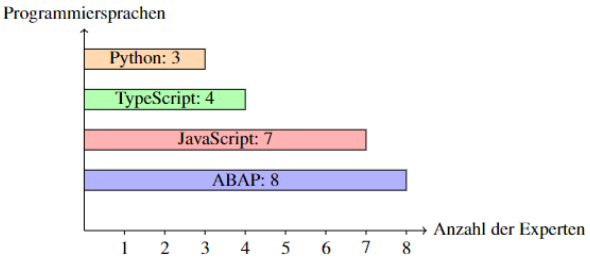


Abbildung 1: Häufigkeit berücksichtigter Programmiersprachen

IDEs
In Bezug auf die verwendeten Integrated Development Environments (IDEs) nutzen sechs Entwickler Eclipse, acht benutzen Visual Studio Code, drei arbeiten mit PyCharm, und ein Experte verwendet Jupyter Notebook. Daher stehen Eclipse, Visual Studio Code, PyCharm und Jupyter Notebook im Mittelpunkt der weiteren Untersuchungen zur KI-unterstützten Softwareentwicklung.

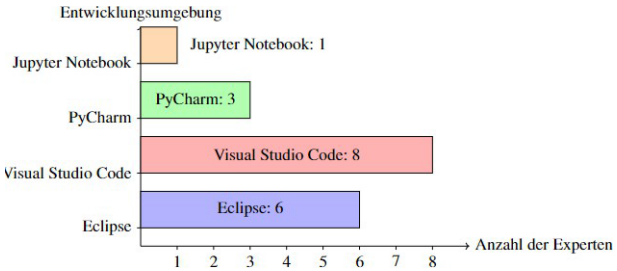


Abbildung 2: Verwendete Programmiersprachen

6. NUTZWERTANALYSE

Im Anschluss an die Experteninterviews wird die Nutzwertanalyse durchgeführt, welche auf den Ergebnissen der Experteninterviews aufbaut. Die Nutzwertanalyse, auch als „Multi-Criteria Decision Analysis (MCDA)“ bekannt, ist eine Methode, die in verschiedenen wissenschaftlichen Disziplinen und in der Praxis eingesetzt wird, um fundierte Entscheidungen zu treffen und alternative Lösungen zu bewerten. Sie ist ein strukturiertes Verfahren zur Bewertung von verschiedenen Alternativen anhand vordefinierter Kriterien. Die Nutzwertanalyse ermöglicht den Vergleich verschiedener Optionen, indem sie objektive Kriterien verwendet, um den Nutzen oder Wert jeder Option zu bestimmen. Dies geschieht durch die Zuweisung numerischer Werte zu den Kriterien und die Aggregation dieser Werte zu einer Gesamtbewertung. [33]
Die Nutzwertanalyse kann entweder als permanentes Tool für langfristige Überwachung oder als einmalige Analyse durchgeführt werden. Diese Abschlussarbeit wählt die zweite Variante, wobei beachtet wird, dass mit zunehmendem technologischem Fortschritt, beziehungsweise neuen Generationen der Vervollständigungstools, die Ergebnisse dieser Projektarbeit veraltet sein können. [8] Im Folgenden Abschnitt werden Teile der im Zuge der Abschlussarbeit durchgeführten Nutzwertanalyse erläutert.

6.1 Festlegung und Definition der Bewertungskriterien

Zunächst werden Kriterien festgelegt, um die Optionen zu bewerten. Diese Kriterien repräsentieren die Anforderungen der T. CON an ein Codevervollständigungstool und werden als Entscheidungsvariablen bezeichnet. Entscheidungsvariablen müssen sinnvoll unterscheidbar und vergleichbar sein. Die Kriterien wurden in Absprache mit dem Betreuer durch Brainstorming, Vorauswahl und Sortierung festgelegt. In dieser Arbeit wurden die Kriterien vom Verfasser definiert und vom Firmenbetreuer genehmigt. Es gibt in der Abschlussarbeit neun Bewertungskriterien:

Anwenderfreundlichkeit: Die Evaluierung der Anwenderfreundlichkeit von KI-Tools erfolgt anhand der Einschätzung ihrer Benutzerfreundlichkeit, welche sich auf die mühelose und intuitive Nutzbarkeit des Tools durch den Anwender bezieht. Hierbei wird geprüft, inwiefern die Softwareanwendung leicht erlernbar ist und ob die Notwendigkeit von Schulungen oder technischem Fachwissen eine Rolle spielt.

Community: Eine Community stellt eine Gruppe von Entwicklern, Benutzern und Experten dar, welche das KI-Tool nutzen oder unterstützen. Zwischen ihnen findet in einem Forum regelmäßiger Austausch statt, um Wissen und Ressourcen zu teilen. Die Community soll bei Fragen und allgemeinen Themen helfen können, zum Beispiel mit Hilfe eines Forums.

Datenschutz: Beim Datenschutzaspekt wird betrachtet, wie gut das zu testende Tool die Sicherheit von sensiblen Informationen gewährleistet. Dies schließt die Speicherung, Verarbeitung und den Zugriff auf die Daten mit ein. Es wird berücksichtigt, ob eingegebene Daten wieder zum Training des KI-Tools verwendet werden.

Geschwindigkeit und Leistung: Die Geschwindigkeit und Leistung eines KI-Tools bezieht sich auf die Reaktionszeit bei der Ausführung von Aufgaben. Es ist wichtig zu bewerten, ob das Tool in Echtzeit arbeitet und ob es komplexe Aufgaben effizient und mit hoher Qualität bewältigen kann, ohne den Programmierfluss zu stören.

Integration in benutzte IDE: Die Integration in benutzte integrierte Entwicklungsumgebungen (IDEs) betrifft die Fähigkeit des KI-Tools, nahtlos in vorhandene Entwicklungsworkflows integriert zu werden. Bei dieser Abschlussarbeit wird auch berücksichtigt, ob die IDEs, in die die Tools integriert werden können, bei der T. CON angewendet werden. Durch die Experteninterviews wissen wir, dass wir die Integration in VS Code, Eclipse, PyCharm sowie in Jupyter Notebook analysieren.

Kosten: Es werden die Kosten pro Zeitabschnitt, wie Monat, Jahr beziehungsweise pro Wort analysiert, die für jeden Benutzer aufkommen. Es wird auch analysiert, welche alternativen Kostenmodelle es gibt.
Qualität der Ergebnisse: Bei den Ergebnissen wird einerseits überprüft, welche Programmiersprachen die Tools unterstützen, aufgrund des Experteninterviews also insbesondere JavaScript und Python.
Des Weiteren ist es wichtig zu prüfen, ob das Tool syntaktische oder semantische Fehler macht und ob die Ergebnisse sinnvoll und konsistent sind. Es wird analysiert, inwiefern Ergebnisse aufeinander aufbauen, oder im Kontext generiert werden.

Support: Der Support für ein KI-Tool umfasst technischen Support, Dokumentationen und Schulungsmaterialien, die bei der Verwendung des Tools unterstützen.

Urheberrecht: Bei der Bewertung urheberrechtlicher Aspekte sind zwei Dinge zu beachten: Wem gehören die im Zuge der Softwareentwicklung generierten Inhalte und wurden beim Training des Sprachmodells urheberrechtliche Verstöße begangen.

6.2 Gewichtung der Kriterien

Jedes Kriterium erhält eine Gewichtung, um seine Bedeutung in der Gesamtbewertung widerzuspiegeln. Dies kann durch Expertenbefragungen oder mithilfe von Analysetechniken erfolgen. Die Gewichtungen zusammengenommen ergeben 100%. Die hohe Anzahl von Aspekten in dieser Nutzwertanalyse, neun an der Zahl, erschwert eine sinnvolle Gewichtung. Bei der sogenannten Nivellierungsverzerrung kommt es durch jedes weitere hinzugefügte Kriterium, welchem ein Gewicht zugeordnet werden muss, eine Bedeutungsverschiebung zuungunsten wichtiger Aspekte. [8] Dass es bei dieser Abschlussarbeit zu Nivellierungsverzerrung kam, zeigt die untenstehende Tabelle. Die niedrigste Gewichtung beträgt 8%, die höchste 14%. Das ist keine große Varianz.
Die Gewichtung der Kriterien ergab sich aus dem Fragebogen, den jeder Experte im Anschluss an das Experteninterview ausfüllen sollte. Bei diesem Fragebogen hat jeder Experte die Wichtigkeit der Aspekte von 0-5 pro Aspekt ausgefüllt. Die Tabelle zeigt, wie die Experten jeden Aspekt hinsichtlich der Wichtigkeit eingeschätzt haben. Diese Zahlen werden anschließend auf 1 (100%) normiert.

Kriterium	Summe	Gewichtet
Anwenderfreundlichkeit	48	11
Community	36	8
Datenschutz	58	13
Geschwindigkeit	46	11
IDE Integration	54	13
Kosten	37	9
Qualität	58	14
Support	42	10
Urheberrecht	49	11
Summe	481	100

Abbildung 3: Gewichtung der Kriterien

6.3 Bewertung der Alternativen

Anwenderfreundlichkeit
GitHub Copilot war das ursprünglich erschienene Tool. Abbildung 5 zeigt das Interface des GitHub Copiloten in einem UI5 Beispielprogramm. Links ist die Chatfunktion zu sehen. Das Tool erkennt, dass in dem Beispiel mit Java-Script programmiert wurde, und schlägt eine Frage hinsichtlich Unittests für JavaScript vor. Rechts ist die Codevervollständigungsfunktion zu sehen. Anhand des Funktionsnamen `onShowHappynewyear` wird berechnet, dass der Anwender einen MessageToast mit der Aufschrift „Happy new Year!“ wünscht. Anhand der Funktionsweisen von Tabnine und Codeium zeigt sich, dass diese Tools stark am GitHub Copilot angelehnt sind, und somit eine vermeintlich bessere Konkurrenz darstellen wollen.
Alle drei Tools zeichnen sich durch intuitive und leicht erlernbare Benutzeroberflächen aus. Der Installationsprozess verläuft für alle drei Tools zügig und unkompliziert. Nach der Installation erscheint jeweils ein Icon, das den Zugang zum entsprechenden Tool ermöglicht.
Den Benutzeroberflächen fehlen jeweils wichtige Informationen zur vollständigen Nutzung der Funktionen, insbesondere zur Navigation durch Codealternativen. Da diese Funktionen leicht erlernt werden können fällt das allerdings nicht weiter ins Gewicht, und alle drei Tools bekommen hinsichtlich Anwenderfreundlichkeit **fünf Punkte**.

Datenschutz
GitHub betont, dass der Copilot derzeit noch nicht zertifiziert ist. Es wird angestrebt, eine SOC 2-Zertifizierung bis Mai 2024 von einem Drittanbieter zertifiziert zu bekommen. Auch betont GitHub, dass der generierte Code nicht zum Training zukünftiger Modelle benutzt wird.[34]
Tabnine benutzt ebenfalls keinen vorgeschlagenen Code zum Training des Modells. Zusätzlich ist es hinsichtlich der Einhaltung der Datenschutzgrundverordnung und nach dem SOC 2-Standard zertifiziert. Das Audit und sämtliche Zertifikate können heruntergeladen werden.[35]
Codeium benutzt ebenfalls keine Codevorschläge, um das Model weiter zu trainieren. Außerdem ist Codeium ebenfalls SOC 2-Zertifiziert. Das Zertifikat ist veröffentlicht, das komplette Audit allerdings nicht. [36]
Zusammengefasst benutzt keines der drei Tools Codevorschläge, um das Model weiter zu trainieren. Insgesamt zeigt sich somit, dass Tabnine und Codeium bereits wichtige Schritte im Bereich Datenschutz vollzogen haben. Besonders positiv hervorzuheben ist, dass Tabnine alle relevanten Informationen hinsichtlich beider Zertifizierungen zur Verfügung stellt. GitHub muss für den Copiloten einiges aufholen, positiv sind allerdings die Bemühungen zur Zertifizierung. Daher wird der Copilot mit **zwei Punkten**, Tabnine mit **fünf Punkten** und Codeium mit **vier Punkten** hinsichtlich des Datenschutzes bewertet.

IDE Integration
Abbildung 4 zeigt, welches Tool in welche der berücksichtigten IDEs integrierbar sind. Codeium ist hier das flexibelste Tool, da es nicht nur in VS Code, Eclipse, PyCharm und Jupyter Notebook integrierbar ist, sondern man das Tool auch vorab im Browser testen kann. Tabnine und der GitHub Copilot sind jeweils in VS Code integrierbar. Der GitHub Copilot zusätzlich in PyCharm, Tabnine zusätzlich in Eclipse.

IDE	GitHub Copilot	Tabnine	Codeium
Browser	Nein	Nein	Ja
VS Code	Ja	Ja	Ja
Eclipse	Nein	Ja	Ja
PyCharm	Ja	Nein	Ja
Jupyter Notebook	Nein	Nein	Ja

Abbildung 4: Integrationsmöglichkeit der Tools in IDEs

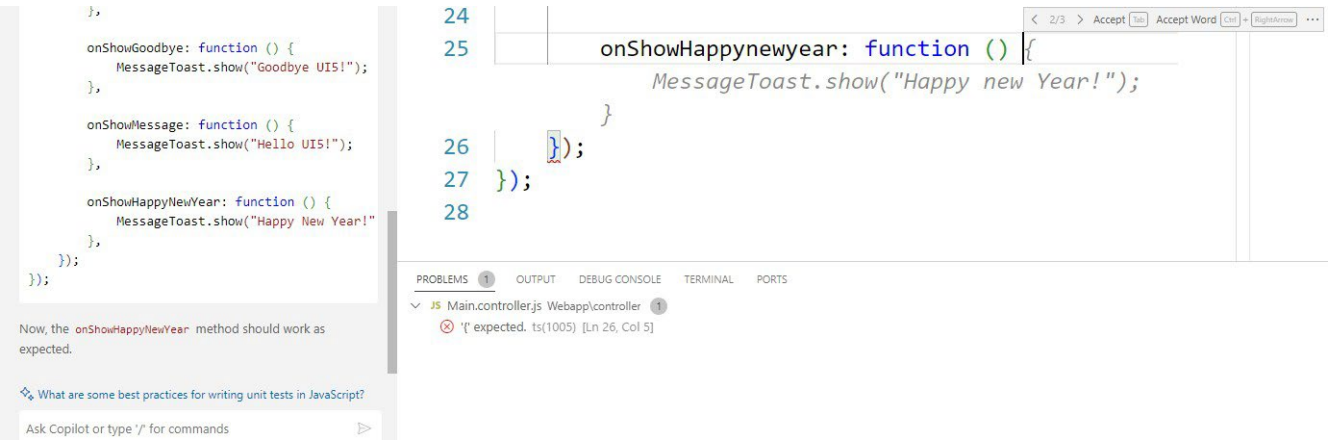


Abbildung 5: Interface des GitHub Copilot

Bei der Bewertung wird berücksichtigt, wie viele Experten welches Tool nutzten. Da VS Code von den meisten Experten benutzt wird, fällt es sehr positiv ins Gewicht, dass alle drei KI-Tools auch eine Integration in diese IDE unterstützen. Dass Tabnine zusätzlich in Eclipse integrierbar ist fällt positiver ins Gewicht als die Integrierbarkeit des GitHub Copilot in PyCharm, da Eclipse mit sechs Experten höher gewichtet ist als PyCharm mit zwei.
In der Gesamtschau erhält der GitHub Copilot daher in dieser Kategorie **drei Punkte**. Tabnine erhält **vier Punkte**, Codeium erhält **fünf Punkte**.

Kosten
Positiv zu bewerten ist bei allen drei Tools, dass es verschiedene Preismodelle gibt, welche den unterschiedlichen Anforderungen, verschiedener Kunden gerecht wird.
Dass das günstigste Preismodell des GitHub Copilot nicht für alle kostenlos ist, sondern 10 US-Dollar im Monat pro Nutzer kostet fällt negativ ins Gewicht. Auch ist negativ anzumerken, dass es „Copilot Enterprise“ nur für Unternehmen gibt, welche die GitHub Enterprise Cloud haben, was die tatsächlichen monatlichen Kosten auf 60 US-Dollar im Monat erhöht. Allerdings bietet das kostengünstigere Preismodell „Copilot Business“ mit 20 US-Dollarn im Monat bereits eine breite Produktpalette, welche für Firmen wichtig sind. Aufgrund der negativen Aspekte gibt es trotz des Modells „Copilot Business“ mit hervorragendem Preisleistungsverhältnis **vier Punkte** im Aspekt Kosten.
Bei Tabnine ist positiv hervorzuheben die kostenlose Version „Starter“, um das Tool auszuprobieren. Allerdings ist das nur ein Preismodell, um die grundlegende Funktion Tabnines zu testen, die 2-3 Wörter, welche vorgeschlagen werden, bieten keine Produktivitätssteigerung. Das zweite Modell „Pro“ für 12 US-Dollar im Monat pro Nutzer ist von den Funktionalitäten vergleichbar mit „Copilot Individual“ und damit teurer als der Copilot. Besonders negativ ins Gewicht fällt, dass Tabnine Unternehmen mit weniger als 100 Softwareentwicklern vom dritten Preismodell „Pro“ abrät, sodass es keine Funktionen wie VPC beziehungsweise finegetunte Modelle gibt.
Außerdem ist die Preisintransparenz, den Preis gibt es nur auf Anfrage, hinsichtlich des dritten Preismodells negativ anzumerken. Aufgrund der aufgezählten Aspekte wird Tabnine hinsichtlich der Kosten mit **drei Punkten** bewertet. Bei Codeium fällt besonders positiv ins Gewicht, dass es die kostenlose Version „Individual“ mit bereits breiter Produktpalette, wie einen chatbasierten KI-Assistenten und Codevervollständigung in Echtzeit gibt. Dass das zweite Preismodell „Teams“ für 12 US-Dollar pro Nutzer und Monat im Jahresabonnement bereits fortgeschrittene Funktionen wie Nutzermanagement und eine personalisierte Codebasis bietet fällt, äußerst positiv ins Gewicht. Codeium ist damit hinsichtlich Preisleistungsverhältnisses das günstigste Tool. Auch bei Codeium gibt es die Preisintransparenz hinsichtlich des Modells „Enterprise“. Da aber wie angemerkt bereits das „Pro“-Modell sehr viele Funktionen beinhaltet, ist die Bewertung von Codeium hinsichtlich der Kosten **fünf Punkte**.

Urheberrecht
Bei der Bewertung der einzelnen Tools hinsichtlich des Urheberrechts wird in zwei Subkategorien aufgeteilt: In der Kategorie **Training des Sprachmodells** wird analysiert, ob beim Training des zugrundeliegenden Modells Urheberrechtsverstöße begangen wurden. In der Kategorie **Rechte am generierten Code** wird darauf eingegangen, welche Rechte die Anbieter der Tools am generierten Code verlangen. In einem folgenden Kapitel wird genauer erläutert, inwiefern das Training des Sprachmodells und die Rechte am generierten Code problematisch sein können.

Training des Sprachmodells:
Tabnine behauptet, dass sein Sprachmodell ausschließlich auf Open-Source-Code trainiert wird, für den das Unternehmen rechtmäßige Lizenzen erworben hat. Welche Trainingsdatensätze benutzt wurden, einschließlich sämtlicher Lizenzen, ist öffentlich zugänglich. Durch diese transparente Vorgehensweise möchte Tabnine nachweisen, dass bei der Modellentwicklung keine Urheberrechtsverletzungen begangen wurden. Die Offenlegung der Trainingsets ermöglicht es einen detaillierten Einblick in die verwendeten Datenquellen zu bekommen, und trägt zur Schaffung von Vertrauen in den Prozess des Modelltrainings bei.

Durch die behauptete konsequente Einhaltung der Lizenzbestimmungen möchte Tabnine seine Verpflichtung zur rechtlich einwandfreien Nutzung von Datenquellen unterstreichen.
Diese Behauptungen sind seitens des Anbieters zwar plausibel vorgetragen worden, dennoch kann nicht mit letzter Sicherheit unabhängig geprüft werden, ob ausschließlich lizensierter Code benutzt wurde. Die Datensätze sind in 36 CSV-Dateien gelistet, wobei jede Datei mehrere 1000 Zeilen beinhaltet, weshalb eine Überprüfung in dieser Abschlussarbeit nicht möglich ist.

Rechte am generierten Code:

In den allgemeinen Geschäftsbedingungen unter Abschnitt 12 betont Tabnine, dass man durch das Erwerben einer Tabninelizenz keinerlei Rechte an den Sprachmodellen erlangt, welche für das Training benutzt wurden. Jedoch wird explizit betont, dass sowohl der entwickelte, als auch der generierte, akzeptierte Code ausschließlich dem Entwickler gehört, und Tabnine auf sämtliche Urheberrechtsansprüche an dem Code oder anderen geistigen Eigentumsansprüche verzichtet, und diese unbefristet selbst genutzt werden können.
In den allgemeinen Geschäftsbedingungen von Tabnine für das kostenpflichtige Preismodell „Teams“ überträgt die Firma im Abschnitt 7.2 dem Nutzer alle Rechte an alles Vorschlägen, die zur Verfügung gestellt, oder zurückgegeben werden. Allerdings betont Tabnine, dass die durch maschinelles Lernen generierten Vorschläge denen anderer Kunden ähneln können, oder sogar übereinstimmen können, sodass dann keine Rechte an den Codevorschlägen gewährt wird.
Das Training des Sprachmodells des GitHub Copilots ist nicht nur Geschäftsgeheimnis, sondern bereits Schauplatz juristischer Auseinandersetzungen geworden. Daher gibt es lediglich **zwei Punkte**. Da Tabnine die Lizenzen zum Training des Sprachmodells sehr detailliert offenlegt, bekommt es in dieser Subkategorie **fünf Punkte**. Codeium verspricht als Alternative zum GitHub Copilot das Modell ausschließlich auf lizensierten Code trainiert zu haben, macht allerdings keine genaueren Angaben hierzu. Daher wird es mit **vier Punkten** bewertet. Da alle drei Tools die Rechte am generierten Code abgeben, wird diese Subkategorie bei allen drei Tools mit **fünf Punkten bewertet**. Dies ergibt aufgerundet einen Gesamtnutzwert hinsichtlich des Urheberrechts von **drei Punkten** für den GitHub Copilot, **fünf Punkte** für Tabnine und **fünf Punkte** für Codeium.

Codequalität

Um die Qualität der drei Codevervollständigungstools zu testen, sind je Tool drei prototypische Anwendungsfälle implementiert worden, welche den Copiloten, Tabnine und Codeium auf Kontextsensitivität, Fähigkeiten im Umfeld von Python, und Fähigkeiten im UI5-Umfeld testen. Für jeden Anwendungsfall sind im Vorfeld konkrete Erwartungen an die Tools formuliert worden, wie sie den Code im Kontext vervollständigen sollen. Nach diesen Erwartungen sind die Tools bewertet worden. Am Ende der Fallstudie wird bewertet, ob die Erwartung erfüllt (e.) oder nicht erfüllt (n.e.) wurde. Alle Anwendungsfälle sind in der Entwicklungsumgebung VS Code durchgeführt worden. Die Prüfung auf Kontextsensitivität war Vorgabe, die Überprüfung auf Python und UI5 ergab sich aus den Experteninterviews.

Kontextsensitivität:

Die Ordnerstruktur des ersten Anwendungsfalls ist in Abbildung 6 zu finden. Bei der Prüfung der Kontextsensitivität wurden in einem Ordner zwei Pythonfiles angelegt.

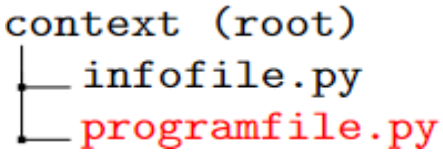


Abbildung 6: Ordnerstruktur Kontextsensitivität

In der Datei `infofile.py` stand Pythonsourcecode. Konkret eine Variable `cptadress`, mit dem Wert `chat.openai.com` initialisiert war. Außerdem eine Funktion, welche für einen Eingabeparameter berechnet, ob er eine Primzahl ist oder nicht.
In der zweiten Datei `programfile.py` wurde unter Benutzung jeweils eines Tools implementiert. Dabei wurde für das jeweilige getestete Tool folgende Erwartungen festgelegt:

1. Zunächst soll die Datei `infofile.py` importiert werden. Spätestens nach dem Tastenanschlag `import my` soll das Codevervollständigungstool erkennen, dass das Infofile zu importieren ist.
2. Anschließend soll eine Variable `my_number` erstellt und mit einem sinnvollen Wert initialisiert werden. Erwartung ist, dass das Tool spätestens nach dem Tastenanschlag `my_num` dies erkennt.
3. Mit Hilfe eines Kommentars, dass die initialisierte Nummer darauf geprüft werden soll, ob sie eine Primzahl ist, soll das Codevervollständigungstool erkennen, dass diese Funktion im Infofile implementiert ist, und diese anwenden.
4. Schließlich wird eine Variable `address` erstellt. Das Codevervollständigungstool soll den Kontext mit der Variable `gptaddress` erkennen, und `address` mit dem gleichen Wert initialisieren.

Anbei die Ergebnisse der drei Codevervollständigungstools:

GitHub Copilot:

1. Bereits nach dem Tastenanschlag von `imp` schlug der Git-Hub Copilot vor, das Infofile zu importieren. (e.)
2. Nach Eingabe von `my_num` vervollständigte der GitHub Copilot die Variable auf `my_number` und initialisierte diese auf den sinnvollen Integerwert 3. (e.)
3. Der Vorschlag, die initialisierte Nummer mit Hilfe der Primzahlfunktion im Infofile zu überprüfen kam automatisch. (e.)
4. Ebenfalls generierte der Copilot einen Printbefehl, in dem stand, dass die Webadresse von gpt `chat.openai.com` ist. (e.)

Dies zeigt, dass der GitHub Copilot hinsichtlich kontext-sensitivität alle Erwartungen erfüllt und übertrifft. Es ist dadurch bewiesen, dass das Tool mindestens den Kontext zwischen zwei Dateien erkennt.

Tabnine:

1. Während des Tastenanschlags wurde bis zum Schluss, als `import myinfofile` bereits eingegeben wurde, kein sinnvoller Vorschlag gemacht. (n.e.)
2. Nach Eingabe von `my_num` wurde auf `my_number` vervollständigt, und diese Variable mit 3 initialisiert, einem sinnvollen Integerwert. (e.)
3. Die Funktion `is_prime` wurde komplett außen vor gelassen. (n.e.)
4. Dagegen vervollständigte Tabnine die Variable `adress = infofile.cptadress(my_number)`. Die String-variable `cpt-adress` wird also aufgerufen. Dies ist syntaktisch unkorrekt, und wirft einen TypeError mit der Fehlermeldung „not callable“. (n.e.)

In der Fallstudie wurde demonstriert, dass Tabnine nicht nur den Kontext nicht korrekt erkennt, sondern auch syntaktisch inkorrekten Code generiert.

Codeium:

1. Codeium hat sofort erkannt, dass das Infofile importiert werden soll. (e.)
2. Die Variable `my_number` wurde per Methodenaufruf der Primzahlfunktion des Infofiles mit Wert 11 initialisiert. Da elf eine Primzahl ist, wurde `my_number` also mit `true` initialisiert. Das ist für eine Variable, deren Name einen Integer- oder Doublewert erwarten lässt also nicht sinnvoll, dennoch syntaktisch korrekt. (n.e.)
3. Damit ist diese Erwartung bereits erfüllt worden. (e.)
4. Der Variable `adress` ist der Wert der Variable `cptadress` des Infofiles zugewiesen. Anschließend wurde die Adresse auf der Konsole ausgegeben. (e.)

Codeium ist somit wie der GitHub Copilot grundsätzlich kontextsensitiv auf mindestens zwei Dateien. Jedoch wird der Kontext teilweise semantisch unkorrekt interpretiert.

Pythonprojekt:

Bei dem Pythonprojekt haben wir einen Ordner mit einer Datei `my_data.csv`. In dieser CSV-Datei sind viele Datensätze. Jeder Datensatz enthält die Variablen `age`, `gender`, `education_level`, `income`, `occupation` und die Zielvariable `target_clomn`. Die Codevervollständigungs-tools sollen anhand der gegebenen Kommentare ein Pythonprojekt generieren, welche Muster innerhalb des Datensatzes suchen. Der Anwendungsfall gliedert sich in zwei Phasen:
In der Datei `data_preprocessing.py` wird der Datensatz in Trainings- und Testdaten aufgeteilt. In der zweiten Datei `train_model.py` wird das Modell mit Hilfe des Trainingssatzes trainiert, evaluiert und gespeichert. Die Ordnerstruktur des Projekts ist in Abbildung 7 zu finden. In den roten Dateien wird implentiert, die blauen Dateien werden durch das geschriebene Pythonskipt generiert.

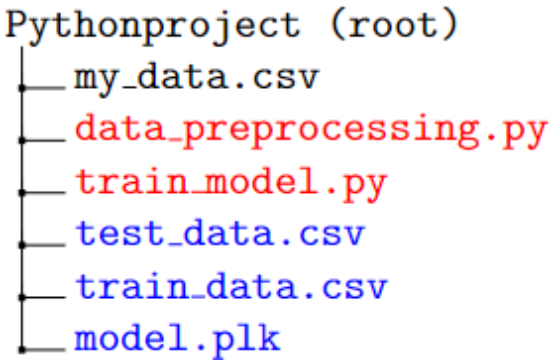


Abbildung 7: Ordnerstruktur des Pythonprojekts

Folgende Erwartungen wurde an das jeweilige Codevervollständigungstool in dem Data Science Projekt gestellt. Die nächsten Schritte ergeben sich aus Python-Kommentaren, die verfasst werden, die Art der Umsetzung aus den importierten Bibliotheken:

1. Da es sich um ein umfangreicheren Anwendungsfall handelt, soll das Codevervollständigungstool objekt-orientiert vorgehen. Es soll also die Funktionen in Methoden schreiben, insbesondere in eine Main-Funktion, und diese aufrufen.
2. Das Programm soll den Datensatz in der Datei `my_data.csv` in Trainings- und Testdaten aufsplitten. Es soll eine sinnvolle Testgröße genommen werden.
3. Bevor das Modell trainiert wird, sollen die nicht numerischen Variablen One-Hot-Codiert werden. Dafür soll das Tool erkennen, welche kategoriale Variablen es in der CSV-Datei gibt. Diese Codierung kategorialer Variablen wird in Data Science Projekten durchgeführt, da Modelle numerischen Input benötigen. [37]
4. Anschließend soll das Modell mit `LinearRegression` trainiert werden. `LinearRegression` ergibt sich aus den importierten Bibliotheken.
5. Die Evaluierung des Modells soll durch die Berechnung des mean squared errors erfolgen.
6. Die Speicherung des Modells soll erfolgen, indem die Funktion `dump` der Bibliothek `joblib` aufgerufen wird.

Hier die Ergebnisse der Tools:

GitHub Copilot:

1. Zu Beginn lieferte der GitHub Copilot Codevorschläge im Skriptstil. Jedoch erkannte das Tool nach der ersten implementierten Funktion, dass auch zukünftig in Funktionen implementiert werden soll. Die Main-Funktion wurde anhand der zuvor implementierten Funktionen in beiden Dateien korrekt umgesetzt. (e.)
2. Die Funktion zum Aufteilen der CSV-Datei in Trainings- und Testdaten ist korrekt umgesetzt worden. Als Testgröße wurde 0.2 gewählt, also 20% des CSV-Datensatzes wird als Testdaten genutzt. Dieser Wert ist Standard. [38] (e.)
3. Die One-Hot-Codierung konnte wurde syntaktisch korrekt umgesetzt. Jedoch der GitHub Copilot aus der CSV-Datei nicht die korrekten kategorialen Variablen auslesen. Diese mussten manuell eingetippt werden. (n.e.)
4. Das Training des Modells wurde anhand der importierten Bibliothek korrekt generiert. (e.)
5. Auch die Evaluierung des Modells wurde korrekt umgesetzt. (e.)
6. Auch die Funktion zum Speichern des Modells wurde korrekt generiert. (e.)

Der GitHub Copilot hat in diesem Data Science Projekt also viele Vorhersagen korrekt getroffen. Außerdem hat sich das Tool an die Bedürfnisse des Entwicklers angepasst, in dem es mit seinen Codevorschlägen zu einer objektorientierten Programmierweise gewechselt hat.

Tabnine:

1. Tabnine hat sich nicht an den objektorientierten Entwicklungsstil angepasst. Die Codevorschläge sind immer wieder in skriptart erstellt worden. Nach Tastenanschlag „def“ ist dann aber immer ein Vorschlag zur Implementierung einer Funktion generiert worden. Die Main-Funktionen konnte das Tool nicht anhand der zuvor implementierten Funktionen generieren. (n.e.)
2. Tabnine hat eine korrekte Funktion generiert, die die Daten in Test- und Trainingsdaten aufteilt. Auch dieses Tool hat die Testgröße auf den Wert 20% gesetzt. (e.)
3. Die One Hot-Codierung wurde korrekt umgesetzt. Jedoch hatte das Tool Probleme die korrekten kategorialen Variablen aus der CSV-Datei zu erkennen. Diese mussten manuell eingefügt werden. (n.e.)
4. Die Funktion zum Training des Modells wurde korrekt umgesetzt. (e.)
5. Auch wurde das Modell korrekt evaluiert. (e.)
6. Die Funktion zum Speichern des Modells wurde ebenfalls korrekt und fehlerfrei generiert. (e.)

Insgesamt zeigt sich, dass Tabnine viel in dem Anwendungsfall korrekt umsetzen konnte. Jedoch wurden die nächsten Schritte, wie Codieren, Trainieren, Evaluieren und Speichern nicht korrekt vorhergesagt, das Tool schlug immer vor, die Daten zu splitten, was, nachdem es bereits umgesetzt wurde nicht mehr sinnvoll war. Außerdem hat Tabnine sich immer an eine skriptweise Implementierung in Python orientiert. Als letztes Manko ist zu erwähnen, dass Tabnine aus den vorhandenen implementierten Funktionen keine fehlerfreie Main-Funktion erstellen konnte.

Codeium:

1. Codeium hat sich ebenfalls nicht an objektorientierter Programmierweise orientiert. Die Codevorschläge änderten sich auch nicht, nachdem bereits manueller Code entwickelt wurde, welcher objektorientierter Programmierweise entspricht. Auch konnte das Programm in der Datei `data_preprocessing.py` die Main-Funktion trotz implementierter Funktionen nicht korrekt umsetzten. In der Datei `train_model.py` wurde sie allerdings korrekt generiert. (e.)
2. Die Funktion zum Aufsplitten des Datensatzes in Trainings- und Testdaten wurde nicht korrekt umgesetzt. Einerseits wurde die Zielvariable nicht korrekt initialisiert. Außerdem wurde keine Testgröße angegeben. Dass die Testgröße 20% ist, musste manuell eingefügt werden. (n.e.)
3. Analog zu den anderen Codevervollständigungstools wurde die One-Hot-Codierung grundsätzlich korrekt umgesetzt. Jedoch hatte auch dieses Tool Probleme, kategoriale Variablen aus der CSV-Datei korrekt auszulesen, weshalb diese manuell eingefügt werden mussten. (n.e.)
4. Die Funktion zum Trainieren des Modells wurde korrekt umgesetzt. (e.)
5. Auch wurde die Funktion zum Evaluieren des Modells anhand des mean squared errors korrekt umgesetzt. (e.)
6. Auch das Speichern des Modells erfolgte fehlerfrei. (e.)

Codeium ist somit in der Lage Pythonprojekte korrekt umzusetzen. Jedoch macht das Tool keine Vorschläge kontextsensitiv zur vorhandenen CSV-Datei. Ebenfalls ändert das Tool seine Codevorschläge nicht hinsichtlich des Programmierstils des Entwicklers und verharrt auf nicht objektorientierter Programmierweise.

UI5-Projekt:

Der dritte Anwendungsfall war ein kleines prototypisches UI5-Projekt. Die Ordnerstruktur des Projektes ist in Abbildung 9 zu finden.

Die Datei `i18n.properties` wird verwendet, um Textressourcen zu speichern. Die Datei enthält Schlüssel-Wert-Paare. Der Schlüssel ist eindeutig, der Wert ist der übersetzte Text. Ein Beispiel in der Anwendung ist der Schlüssel `homePageTitel` mit dem Wert „UI5 Walkthrough“. In der Datei `manifest.json` wird die UI5 Anwendung zentral konfiguriert. Es werden vor allem Abhängigkeiten definiert. Abhängigkeiten in diesem Usecase waren konkret die Pfade zur Datei `i18n.properties`, den CSS, View und Controllerelementen. Die Tools wurden an den rot markierten Dateien `HelloPanel.controller.js` und `component.js` getestet. In der Datei `component.js` wird das UI5 Modell initialisiert. Dabei wird auf die Datei `manifest.json` verwiesen. In der Initialisierungsfunktion wird die Variable `oData`, ein JavaScript-Objekt mit UI5-Daten befüllt.

Die Datei `HelloPanel.controller.js` ist eine Controller-Datei, welche implementiert, wie die Webanwendung beim Klick auf verschiedene HTML-Elemente reagieren soll.

In der Datei `HelloPanel.view.xml` wurden drei Buttons implementiert, die beim Klick das ausführen sollen, was in den Funktionen `onShowHello`, `onOpenDialog` und `onCloseDialog` implementiert wird. Diese drei Funktionen werden in `HelloPanel.controller.js` implementiert. Daher ergeben sich folgende Erwartungen an die Codevervollständigungstools:

```
sap.ui.demo.walkthrough.webapp (root)
├── controller
│   ├── App.controller.js
│   └── HelloPanel.controller.js
├── css
│   └── style.css
├── i18n
│   └── i18n.properties
├── view
│   ├── App.view.xml
│   ├── HelloDialog.fragment.xml
│   └── HelloPanel.view.xml
├── component.js
├── index.html
├── index.js
└── manifest.json
```

Abbildung 8: Ordnerstruktur des UI5-Projekts

1. In beiden Dateien soll der Modus „use strict“ verwendet werden. Dies ist eine Sicherheitsmaßnahme, die das Programm nur kompilieren lässt, wenn spezielle Programmierkonventionen, wie das nicht Verwenden globaler Variablen eingehalten wird. [39]
2. In der Datei `component.js` wird auf `manifest.json` verwiesen und in der Initialisierungsfunktion die Variable `oData` mit UI5-Daten befüllt.
3. In der Datei `HelloPanel.controller.js` wird erkannt, welche drei Funktionen sich aus dem zugehörigen View-Element `HelloPanel.View.xml` ableiten lassen.
4. In beiden Dateien wird dem Controller mitgeteilt, um welche Datei es sich handelt. Dabei muss der Pfad zur Datei korrekt angegeben werden.

GitHub Copilot:

1. "Use strict" ist korrekt generiert worden. (e.)
2. In der Datei `component.js` wurde die Datei `manifest.json` korrekt eingebunden. Außerdem wurde die `oData`-Variable korrekt mit UI5-Daten befüllt. (e.)
3. Der GitHub Copilot hat korrekt erkannt, welche Funktionen sich aus dem View-Element ableiten lassen. Bei der Funktion `onShowHello` wurde sogar das Key-Value-Paar `HelloMsg` aus der Datei `i18n.properties` benutzt, und sinnvoll in die Implementierung eingefügt. (e.)
4. In beiden Dateien wurde der Pfad zur Datei korrekt angegeben. (e.)

Der GitHub Copilot konnte also in dem Anwendungsfall mit Hilfe seiner Codevorschläge alle Erwartungen erfüllen. Besonders hervorzuheben ist, dass bei der Implementierung der Funktion `onShowHello` der semantisch sinnvolle Kontext zu einem Key-Value-Paar in den i18n-Properties hergestellt wurde.

Tabnine:

1. Tabnine hat "use strict“ in der ersten Datei `component.js` nach Tastenanschlag "u" vervollständigt. In der zweiten Datei wurde "use strict “ automatisch vervollständigt. (e.)
2. Tabnine hat die Notwendigkeit zur Implementierung einer Init-Funktion zwar erkannt, jedoch waren die Codevorschläge in mehrerer Hinsicht syntaktisch inkorrekt. Es wurde noch die Initialisierung einer oData-Variable korrekt vorgeschlagen. Jedoch wurde dieser nicht der Wert "UI“, sondern der sinnlose Wert "John “ zugewiesen. Außerdem erstellte das Programm kein neues JSONModel, mit der oData-Variable als Input. (n.e.)
3. Tabnine konnte nicht erkennen, welche Funktionen der zugehörigen View zu implementieren sind. Es wurden Funktionen vorgeschlagen wie `onAfterRendering`, welche in unserem Anwendungsfall nicht existieren. Nach manueller Vorgabe, die Funktion `onShowHello` zu implementieren kam ein Codevorschlag "HelloWorld“ auf der Konsole auszugeben. Das ist semantisch korrekt, jedoch nicht zielführend. Nach fertiger Implementierung der Funktion schlug Tabnine wiederholt vor die Funktion `onShowHello` wiederholt zu implementieren. (n.e.)
4. Der Pfad, auf den referenziert werden muss, wurde in den Dateien korrekt erkannt. (e.)

Tabnine machte bei der Initialisierung der UI-Komponente syntaktische Fehler. Außerdem konnte das Tool nicht aus dem Kontext heraus erkennen, welche Funktionen im Hello-Panel zu implementieren sind.

Codeium:

1. Codeium hat „use strict“ zweimal korrekt vorgeschlagen. (e.)
2. Codeium hat die Initfunktion generiert, jedoch einen syntaktischen Fehler eingebaut. Der oData-Variable wäre der sinnlose Wert „World“ statt „UI5“ zugewiesen worden. (n.e.)
3. Auch Codeium hat aus dem Viewelement heraus erkannt, welche drei Funktionen zu implementieren sind, und die Implementierung syntaktisch korrekt umgesetzt. (e.)
4. Codeium erkannte in beiden Dateien den Pfad, auf den zu verweisen war. (e.)

Codeium machte in diesem Anwendungsfall Fehler bei der Initialisierung der UI-Komponente. Dennoch konnte das Tools aus dem Kontext heraus erkennen, welche Funktionen zu implementieren sind, und um welche Dateienkomponente es sich jeweils handelt.

Für die Kontextsensitivität konnte ein Punkt erreicht werden. Im Pythonprojekt und im UI5-Projekt jeweils zwei. Anbei steht für jede Bewertung auch die Bewertung der einzelnen Unterkategorien. Der GitHub Copilot wurde mit fünf Punkten (1+2+2), Tabnine mit zwei Punkten (0+1+1) und Codeium mit vier Punkten (1+2+1) in Bezug auf Codequalität bewertet.

		GitHubCopilot		Tabnine		Codeium	
Kategorie	Gewicht	ung.	gew.	ung.	gew.	ung.	gew.
Anwenderfreundl.	11	5	55	5	55	5	55
Community	8	4	32	0	0	4	32
Datenschutz	13	2	26	5	65	4	52
Geschwindigkeit	11	5	55	2	22	5	55
IDE Integration	13	3	39	4	52	5	65
Kosten	9	4	36	3	27	5	45
Qualität	14	5	70	2	28	4	56
Support	10	3	30	5	50	5	50
Urheberrecht	11	4	44	5	55	3	33
Summe	100	35	387	31	354	40	443

Abbildung 9: Tabelle mit den Gesamtnutzwerten

6.4 Berechnung des Gesamtnutzwertes

Die Bewertungen der Einzelkriterien werden mit ihren Gewichtungen multipliziert und summiert (siehe Abbildung 9). Der höchstmögliche Nutzwert liegt bei 500 Punkten. Codeium kommt auf einen Nutzwert von 443 Punkten. Der Copilot auf 387 Punkte und Tabnine auf 354 Punkte. Der Copilot überzeugt durch hohe Qualität in den Ergebnissen der Codegenerierung und ist sehr anwenderfreundlich, muss jedoch im Bereich Datenschutzzertifizierung noch nachholen. Außerdem wäre es schön, wenn er in mehr IDEs integrierbar wäre. Tabnine hält sich zwar an hohe Urheberrechts-anforderungen, kann allerdings insbesondere nicht in der Kategorie Codequalität punkten. Codeium überzeugt mit guter Codequalität, wenn auch etwas schwächer als der GitHub Copilot. In der Kategorie IDE-Integration sticht er hervor.

6.5 Gewichtungsproblem

In dieser Nutzwertanalyse wurde die Gewichtung anhand der Einschätzungen der Experten in den Experteninterviews festgelegt. Jedoch kam es dabei zu Verzerrungen, da sich die Experten relativ uneinig waren. Daher wurde noch einmal mit Alternativen Gewichten berechnet: Anwenderfreundlichkeit 10%, Community 2,5%, Datenschutz 20%, Geschwindigkeit 15%, IDE Integration 10%, Kosten 5%, Qualität 25%, Support 2,5% und Urheberrecht 10%. Die Bewertungen bleiben gleich. Auch mit dieser sinnvolleren Gewichtung hat Codeium mit 435,5 Punkten das höchste Gewicht, gefolgt vom Copiloten mit 397,5 Punkten und Tabnine mit 347,5 Punkten. Das Ergebnis, dass Codeium also den größten Nutzwert hat hält einer sinnvolleren Gewichtung stand.

7. FAZIT

7.1 Kernergebnisse

Nach den Experteninterviews und der Nutzwertanalyse lautet das Kernergebnis, dass Codeium und der GitHub Copilot sich anhand der Bedürfnisse der T. CON eignen, im Softwareentwicklungsprozess eingesetzt zu werden.
Der GitHub Copilot viel besonders positiv auf im Aspekt der Codequalität. Er nahm bei allen drei Tests, mit denen die Qualität evaluiert wurde, Bezug auf den Gesamt-kontext des Projekts, und machte semantisch und syntaktische Codevorschläge. Lediglich beim Auslesen einer CSV-Datei war der Codevorschlag nicht korrekt zur Laufzeit. GitHub Copilot ist in VS Code und PyCharm integrierbar. Der in den Prompt eingegebene Code wird nicht zum Training zukünftiger Generationen des Sprachmodells benutzt, und GitHub erhebt keinen urheberrechtlichen Anspruch auf den generierten Code. Es ist jedoch negativ anzumerken, dass unabhängige Datenschutzzertifizierungen noch fehlen.
Die größte Stärke des Tools Codeium ist seine flexible Einsetzbarkeit in alle bei der T. CON genutzten IDEs. Das sind VS Code, Eclipse, PyCharm und Jupyter Notebook. Die Codequalität konnte nicht an die des GitHub Copiloten heranreichen. Beim Testen der Qualität fiel er durch semantische Fehler negativ auf. So wollte er eine Variable mit dem Namen my_numer, was auf einen Integer oder Doublewert schließen lässt mit einem Booleanwert initialisieren. Auch dieses Tool erzeugte einen Laufzeitfehler beim Auslesen der CSV-Datei. Auch bei Codeium wird der in den Prompt ein-gegebene Code nicht zum Training des Modells verwendet. Im Gegensatz zum GitHub Copilot wurde Codeium SOC 2-Zertifiziert.

7.2 Güte der Lösung

Diese Projektarbeit hatte zum Ziel, drei Codevervollständigungstools anhand definierter Kriterien zu evaluieren. Folgende quantitative Kennzahlen bewerten die Lösung:
Der GitHub Copilot kostet 19 US-Dollar im Monat pro Benutzer. Codeium kostet 12 US-Dollar im Monat pro Benutzer.
Im Zuge der Qualitätstests wurden insgesamt 14 Erwartungen an die Tools definiert. Diese Erwartungen bezogen sich auf syntaktisch und zur Laufzeit korrekte, sowie semantisch sinnvolle Codevorschläge. Vier Erwartungen bei der Analyse der Kontextsensitivität, sechs bei einem kleinen Pythonprojekt und vier bei einem Projekt im UI5-Umfeld. Der GitHub Copilot konnte insgesamt 13 (4+5+4) dieser Erwartungen erfüllen. Codeium konnte zehn (3+4+3) der 14 Erwartungen erfüllen.
Bei der Nutzwertanalyse wurde der GitHub Copilot nach Gewichtung ohne Nivellierungsverzerrung mit 397,5 aus 500 Punkten bewertet, das sind 79,5%. Codeium kam auf 435,5 aus 500 Punkten, also 87,1%. Die Güte der Lösung kann auch anhand nicht quantitativer Kennzahlen erhoben werden. In der Abschlussarbeit gibt es Einschränkungen hinsichtlich der erarbeiteten Lösungen, dennoch konnten alle Ziele umgesetzt werden.
Die Experteninterviews waren sehr zeitaufwendig, daher konnte nur ein kleiner Querschnitt der T. CON interviewt werden. Die interviewten Experten haben ohne Ausnahme fachlich wertvolle Antworten gegeben, welche für die Nutzwertanalyse eine gute Basis bildeten.
Die Nutzwertanalyse konnte wie in der Zielsetzung formuliert umgesetzt werden. Die theoretischen Aspekte der Nutzwertanalyse konnten gut abgearbeitet werden, jedoch war man häufig auf Angaben der Unternehmen angewiesen. Die Qualität der Tools wurde durch die Implementierung umfangreicher praktischer Usecases bewertet, welche allerdings prototypischer Natur waren, und stark abstrahiert werden mussten. Es konnte begründet werden, warum sich der GitHub Copilot und Codeium eignen, Tabnine dagegen nicht. Es war ursprünglich geplant, die Gewichtung in der Nutzwertanalyse anhand der Expertenbefragung auszurichten. Dies stellte sich allerdings als nicht sinnvoll heraus, da es zur Nivellierungsverzerrung kam. Wichtige Muss-Aspekte wie Datenschutz und Codequalität waren dadurch nur unwesentlich höher gewichtet als Aspekte wie Community, welche eine Kann-Anforderung darstellen.

7.3 Handlungsempfehlung

Es wird ein offener Umgang der Firma mit den Tools Codedium und GitHub Copilot empfohlen. Der Copilot hat in den Analysen tendenziell eine höhere Codequalität als Codeium, jedoch entwickeln sich die Sprachmodelle beider Tools kontinuierlich weiter. Insbesondere die Entwickler, welche Interesse an den beiden Tools zeigen, sollten diese benutzen dürfen. So kann weitere Erfahrung gesammelt werden. Außerdem sollten, sobald die SAP eigene Tools veröffentlicht, auch diese erforscht werden.

7.4 Zukunftsausblick

Laut Gartner befindet sich generative KI derzeit auf dem Gipfel überhöhter Erwartungen im Hype Cycle for Emerging Technologies. Generative KI wird als Teil eines breiten Trends betrachtet, der neue Möglichkeiten für Innovationen eröffnet. Es wird prognostiziert, dass diese Technologien innerhalb der nächsten zwei bis fünf Jahren bedeutenden Nutzen bringen werden. Jedoch warnt das Institut alle Führungskräfte, dass ihre Aufmerksamkeit auch auf andere aufkommende Technologien wie Clouddienste, sowie Sicherheit und Datenschutz richten müssen. Technologien zu benutzen, welche sich in einem frühen Stadium des Hype Cycles befinden, bergen höhere Risiken, aber auch potenziell größere Vorteile [40].
Die SAP hat auf der TechEd 2023 viel angekündigt, jedoch bis dato wenig veröffentlicht. Jeder SAP-Entwickler soll laut dem Konzern in naher Zukunft mit generativer KI arbeiten. Die SAP stellte ein generatives KI-Modell vor, welches in SAP Build Code integriert ist, und stark dem Interface des GitHub Copilot ähnelt [41]. Außerdem soll die HANA Cloud an Vektordatenbanken angepasst werden, welche die Interaktion von LLMs und geschäftskritischen Daten verbessern soll. Außerdem baut die SAP Lernangebote aus, welche Einfluss auf die rollenbasierte Zertifizierung hat, was für den Partnerstatus der T. CON relevant ist [2]. Die SAP verspricht insgesamt in den nächsten zwei Jahren eine „dreistellige Anzahl von Neuerungen“ auf den Markt zu bringen. [42]

8 LITERATUR

[1] Holger Schmidt. KI macht Softwareentwickler 30 bis 50 Prozent produktiver, 2024. zuletzt besichtigt: 15. März 2024.
[2] Bernhard Luck. SAP macht jeden Entwickler zum Entwickler für generative KI, 2023. zuletzt besichtigt: 06. Januar 2024.
[3] IDC. IDC: Die ICT-Ausgaben in der DACH-Region werden in diesem Jahr 275 Milliarden US-Dollar erreichen, der Investitionsschwerpunkt liegt auf KI, 2023. zuletzt besichtigt: 06. Januar 2024.
[4] Hilker. Generative KI am Zenit: Gartners Hype-Zyklus 2023, 2023. zuletzt besichtigt: 06. Januar 2024.
[5] Andreas Streim. KI gilt in der deutschen Wirtschaft als Zukunftstechnologie – wird aber selten genutzt, 2023. zuletzt besichtigt: 06. Januar 2024.
[6] Tcon. Mehrwert durch Softwareerfahrung, Beratung und Service, 2023. zuletzt besichtigt: 15. Juni 2022.
[7] Robert Kaiser. Qualitative Experteninterviews: Konzeptionelle Grundlagen und praktische Durchführung. Springer Fachmedien Wiesbaden, 2014.
[8] Jörg B. Kühnapfel. Scoring und Nutzwertanalysen: Ein Leitfaden für die Praxis. Springer Fachmedien Wiesbaden, 2021.
[9] Johannes Behrndt. SAPUI5 OpenUI5, 2022. zuletzt besichtigt: 06. Januar 2024.
[10] Ingo Biermann. SAPUI5, 2021. zuletzt besichtigt: 06. Januar 2024.

[11] Jann Raveling. Was ist künstliche Intelligenz?, 2023. zuletzt besichtigt: 30. Januar 2024.
[12] Alexander Kästner, Maren Bührig, Janina Holm, Dominik Klee, Michael Löbbert, Marcel Scherbinek, and Vincent Schmid. SAP Data Intelligence. December 2020.
[13] Thimira Amaratunga. NLP Through the Ages, page 9–54. Apress, 2023.
[14] Stefan Luber. Was ist ein Large Language Model (LLM)?, 2023. zuletzt besichtigt: 30. Januar 2024.
[15] Data Guard. SOC 2 oder ISO 27001 Zertifizierung: Vergleich der InfoSec-Standards, 2023. zuletzt besichtigt: 06. Januar 2024.
[16] Andrew Williams. Was bedeutet SOC 2-Konformität?, 2022. zuletzt besichtigt: 06. Januar 2024.
[17] Sophie Suske, Sören Siebert. DSGVO: Was sollten Webseitenbetreiber und Unternehmer über die Datenschutz-Grundverordnung wissen?, 2021. zuletzt besichtigt: 06. Januar 2024.
[18] Craig Smith. Hallucinations could blunt ChatGPT's success, 2023. zuletzt besichtigt: 06. Januar 2024.
[19] Christian Meier. Warum die KI so gerne lügt, 2023. zuletzt besichtigt: 06. Januar 2024.
[20] Gerrit De Vynck. ChatGPT 'hallucinates': some researchers worry it isn't fixable., 2023. zuletzt besichtigt: 06. Januar 2024.
[21] Erich Moechel. Künftige KI-Modelle potenziell von Demenz bedroht, 2023. zuletzt besichtigt: 06. Januar 2024.
[22] Felix Holtermann. Wird ChatGPT dümmer? Das sagt eine Stanford-Studie, 2023. zuletzt besichtigt: 06. Januar 2024.
[23] Peter Zellinger. ChatGPT wird immer dümmer, doch niemand weiß, warum, 2023. zuletzt besichtigt: 06. Januar 2024.
[24] Holger Schmidt, Peter Buxmann. Matthias Orthwein: "Urheberrecht für KI-Inhalte wird ein Problem für die Softwareindustrie", 2024. zuletzt besichtigt: 06. Januar 2024.
[25] Charlotte Voß. New York Times klagt gegen Microsoft und OpenAI, 2023. zuletzt besichtigt: 06. Januar 2024.
[26] Julia Bald. Mithilfe künstlicher Intelligenz plötzlich Urheber?, 2023. zuletzt besichtigt: 06. Januar 2024.
[27] Assecor. GitHub Copilot, 2023. zuletzt besichtigt: 06. Januar 2024.
[28] IONOS. GitHub Copilot: Der Programmierassistent im Überblick, 2023. zuletzt besichtigt: 06. Januar 2024.
[29] Gedeon Rauch. Wie funktioniert GitHub Copilot?, 2023. zuletzt besichtigt: 06. Januar 2024.
[30] Lars Becker. Was ist Tabnine? Alle Infos über den Code-Assistenten, 2023. zuletzt besichtigt: 06. Januar 2024.
[31] Gedeon Rauch. Wie funktioniert Tabnine?, 2023. zuletzt besichtigt: 06. Januar 2024.
[32] Finn Hillebrandt. Codeium, 2023. zuletzt besichtigt: 06. Januar 2024.
[33] Philipp Steubel. Die Nutzwertanalyse: Definition, Anwendung und Beispiele!, 2023. zuletzt besichtigt: 06. Januar 2024.
[34] GitHub. Github Copilot Trust Center, 2024. zuletzt besichtigt: 30. Januar 2024.
[35] Tabnine. Trust Center, 2024. zuletzt besichtigt: 30. Januar 2024.
[36] Codeium. Codeium is SOC2 compliant, 2023. zuletzt besichtigt: 30. Januar 2024.
[37] Wojtek Fulmyk. One-Hot Encoding — A Brief Explanation, 2023. zuletzt besichtigt: 15. März 2024.
[38] Sajid Lhessani. What is the difference between training and test dataset?, 2022. zuletzt besichtigt: 15. März 2024.
[39] John Resig. ECMAScript 5 Strict Mode, JSON, and More, 2009. zuletzt besichtigt: 15. März 2024.
[40] Gartner. Gartner places generative ai on the peak of inflated expectations on the 2023 hype cycle for emerging technologies, 2023. zuletzt besichtigt: 06. Januar 2024.
[41] JG Chirapurath. Supercharging developer productivity with sap build code, 2023. zuletzt besichtigt: 06. Januar 2024.
[42] Christof Kerkmann. Warum SAP bisher mehr ankündigt als liefert, 2024. zuletzt besichtigt: 06. Januar 2024.

Entwicklung einer integrierten Microservice-Architektur am Beispiel von modularisierten RPA-Prozessen

Max-Arthur Klink

Hochschule Pforzheim
Tiefenbronnerstr. 65
75175 Pforzheim
klinkmax@hs-pforzheim.de

Frank Morelli

Hochschule Pforzheim
Tiefenbronner Straße 65
75175 Pforzheim
frank.morelli@hs-pforzheim.de

ABSTRACT
Die vorliegende Ausarbeitung untersucht die Entwicklung einer modularen Microservice-Architektur zur Optimierung von Robotic Process Automation (RPA). Ziel ist es, Flexibilität, Skalierbarkeit und Wartbarkeit zu verbessern, indem monolithische RPA-Prozesse in unabhängige, wiederverwendbare Microservices aufgeteilt werden. Ein praxisnahes Implementierungsmodell adressiert dabei zentrale Anforderungen wie Modularität, lose Kopplung und Resilienz. Die Differenzierung zwischen wertgenerierenden und unterstützenden Microservices ermöglicht eine effiziente Prozessgestaltung, während ein zentraler Katalog und Orchestrierungswerkzeuge die Verwaltung und Integration erleichtern. Experteninterviews lieferten fundierte Einblicke in die Priorisierung relevanter Architekturmerkmale. Die Ergebnisse zeigen, dass die entwickelte Architektur wesentliche Effizienzgewinne und eine erhöhte Anpassungsfähigkeit ermöglicht. Abschließend wird die Architektur hinsichtlich zentraler Mehrwerte evaluiert, und es werden konkrete Handlungsempfehlungen für zukünftige Anwendungen und Erweiterungen gegeben.

SCHLÜSSELWÖRTER
Microservices; Robotic Process Automation (RPA); Architektur; Modularisierung; Prozessautomatisierung; Architekturprinzipien; Implementierung.

EINLEITUNG

Die Automatisierung von Geschäftsprozessen durch den Einsatz von Robotic Process Automation (RPA) ist ein zentraler und stetig wachsender Bereich in der IT-Strategie vieler Unternehmen. Durch die Verwendung dieser Technologie können standardisierbare Prozesse, welche in der Komplexität variieren, automatisiert werden. Aktuell erfolgt die Analyse und Automatisierung von Geschäftsprozessen jeweils getrennt voneinander. Allerdings treten bei einer Vielzahl von Geschäftsprozessen Ähnlichkeiten im Hinblick auf Inhalte und Struktur der darunter liegenden Teilprozesse auf. Dies führt häufig dazu, dass man innerhalb dieser Prozesse ähnliche Funktionen bzw. gleiche Bestandteile verwendet. Die Automatisierung von Geschäftsprozessen verfolgt in diesem Fall einen monolithischen Ansatz. Dies bedeutet, dass die gesamte Logik und Funktionalität innerhalb jedes einzelnen automatisierten Prozesses integriert ist.
Die monolithische Struktur der aktuellen RPA-Prozesse bringt verschiedene Herausforderungen mit sich, insbesondere hinsichtlich der Prozesswiederverwendbarkeit und -wartbarkeit. Steigende oder sich ändernde Anforderungen an die Automatisierung erfordern oft Anpassungen mehrere Komponenten innerhalb eines Prozesses, wodurch die Aktualisierung komplex und zeitaufwändig ist. Änderungen an einzelnen Bestandteilen können mehrere Prozesse betreffen, wodurch diese ebenfalls geändert werden müssen, was zusätzlich die Wartbarkeit und Flexibilität der Prozesse einschränkt. Dies zeigt den Bedarf nach einer effizienteren Lösung für die Konzipierung von RPA-Prozessen auf.
Durch die Unterteillung der einzelnen Bestandteile der Prozesse in unabhängige Microservices lassen sich zukünftig Funktionen und Komponenten separat entwickeln und bereitstellen. Damit soll eine einfachere Wartung und Aktualisierung der gesamten Prozesslandschaft erzielt werden, da man Änderungen nur noch in den betroffenen Microservices vornehmen muss und nicht in jedem einzelnen automatisierten Prozess. Die Zielsetzung des vorliegenden Artikels besteht in der Veranschaulichung einer Microservice-Architektur am Beispiel von modularisierten RPA-Prozessen. Der vorliegende Artikel untersucht die Thematik aus folgender Forschungsperspektive:

1. Wie lassen sich Microservices katalogisieren?
2. Wie kann eine Microservice-Architektur im RPA-Umfeld ausgestaltet werden?

ROBOTIC PROCESS AUTOMATION

Definition und Grundlagen

Bei Robotic Process Automation handelt es sich um eine Art der Prozessautomatisierung. Unter der Prozessautomatisierung wird der Einsatz von Software und Technologien zur Automatisierung von Geschäftsprozessen verstanden. Das Ziel der Prozessautomatisierung ist die Erreichung der Geschäftsziele, die Verbesserung der Rentabilität und der Wettbewerbsfähigkeit der Unternehmen (SAP o. D.). Der Begriff RPA kam erstmals im Jahr 2000 auf, fand jedoch bis zum Jahr 2012 kaum Verwendung und erlangte erst dann an Bedeutung (Doguc 2020, S. 470 f.). Im Herbst 2015 befand sich RPA in der Phase der frühen Mehrheit, was bedeutet, dass die Technologie bereits von einer Vielzahl von Unternehmen eingesetzt wird

(Willcocks et al. 2015, S. 3; Karnowski 2013, S. 520). Mit dem Verlauf der Jahre konnte ein signifikantes Wachstum in diesem Bereich beobachtet werden (Doguc 2020, S. 471). Es gibt eine Vielzahl an Definitionen für RPA. Der vorliegende Artikel basiert auf folgender Definition: „Bei RPA handelt es sich um eine Technologie, die es Unternehmen ermöglicht, repetitive, zeitaufwändige und regelbasierte Aufgaben zu automatisieren, wodurch menschliche Mitarbeiter entlastet und die Effizienz gesteigert wird.“ RPA trägt so zur Verbesserung der Geschäftsprozesse und zur Erreichung strategischer Unternehmensziele bei. Bei RPA handelt es sich um ein Software-Programm mit dem Softwareroboter programmiert werden können (Institute for Robotic Process Automation & Artificial Intelligence o. D.; Gartner o. D.b; PWC South Africa o. D.; Langmann und Turi 2021, S. 6). Diese Softwareroboter interagieren dabei mit der Präsentationsschicht anderer Programme. Sie verhalten sich dabei wie ein Mensch gegenüber der grafischen Benutzeroberfläche des Systems, ohne andere Schichten zu verwenden (Willcocks et al. 2015, S. 7 f.; van der Aalst et al. 2018, S. 269). Moderne RPA-Lösungen erweitern zusätzlich ihren Anwendungsbereich, indem sie neben der grafischen Oberflächenautomatisierung auch die Integration von API-Aufrufen ermöglichen. Durch die Verknüpfung mit Backend-Systemen über APIs lassen sich sowohl Front-End- als auch Back-End-Automatisierungen realisieren. Eine Studie hat zudem gezeigt, dass die Verwendung von APIs im RPA-Kontext gegenüber der reinen GUI-Automatisierung zu empfehlen ist, da unter anderem die Ausführungsgeschwindigkeit gesteigert werden kann (Průcha und Skrbek 2022, S. 260-262, 264, 267-272; AWS o. D.).

Der Abbildung 1 kann das RPA-Schichtmodell entnommen werden. Entsprechend der in der Literatur dargestellten Ansichten operiert RPA im Gegensatz zu anderen Automatisierungslösungen prinzipiell auf der Präsentationsschicht. Es ist jedoch anzumerken, dass insbesondere neuere RPA-Systeme erweiterte Funktionen bieten, die auch die Interaktion mit anderen Schichten, wie der Geschäftslogik- und Datenebene, ermöglichen (Drawehn et al. 2022, S. 14).

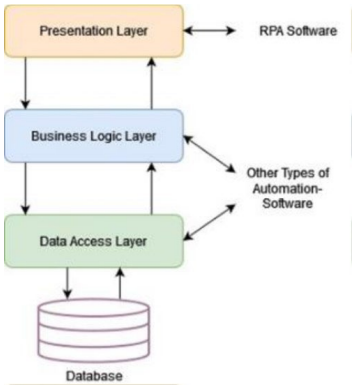


Abbildung 1: RPA-Schichtmodell Quelle: Eigene Darstellung in Anlehnung an Dey und Das 2019, S. 222; Drawehn et al. 2022, S. 14

Indem die RPA-Software auf den bestehenden Systemen aufsetzt, kann es auf der vorhandenen Infrastruktur implementiert werden, ohne dass Änderungen am IT-Backend erforderlich sind (Willcocks et al. 2015, S. 7; Berruti et al. 2017). Daraus folgt, dass RPA auf einen sogenannten „outside-in“ Ansatz setzt, da das Informationssystem unverändert bleibt. In einem klassischen „inside-out“ Ansatz, wären Änderungen am Informationssystem die Folge (van der Aalst et al. 2018, S. 269, 271). Weiterhin lässt sich RPA dem "Lightweight IT“-Konzept zuordnen: Hierunter fallen IT-Systeme, die unkompliziert auf die Bedürfnisse erfahrener Nutzer reagieren, ohne auf komplexe und umfassende IT-Infrastrukturen angewiesen zu sein (Bygstad 2017, S. 182; Willcocks et al. 2015, S. 21 f.). Innerhalb von RPA kann man zwischen drei Typen unterscheiden: Attended, Unattended und Hybrides RPA (Axmann und Harmoko 2020, S. 559).

Bei Attended RPA, auch als Robotic Desktop Automation (RDA) bezeichnet, kann ein Software-Roboter direkt auf dem Desktop des Benutzers ausgeführt werden. Der Benutzer ist in der Lage den Roboter zu starten, zu überwachen und mit ihm über einen Bildschirm zu interagieren. Der Roboter seinerseits kann mit verschiedenen Anwendungen interagieren wodurch sich verschiedene Arbeitsschritte automatisieren lassen. Attended RPA fungiert wie ein persönlicher Assistent, weil es bestimmte Aufgaben übernimmt und ausführt. Ein Nachteil von Attended RPA besteht darin, dass der Roboter den Computer des Users benötigt. Während der Programmlaufzeit ist dieser dann nicht mehr in der Lage, seinen Computer anderweitig zu verwenden (Langmann und Turi 2021, S. 6; Axmann und Harmoko 2020, S. 559). Eine Untersuchung von Anwendungsszenarien ergibt, dass Atended RPA primär bei Prozessen zum Einsatz kommt, die nicht komplett regelbasiert automatisierbar sind bzw. an verschiedenen Stellen menschliche Entscheidungen benötigen. Ein Anwendungsfall für Attended RPA ist z.B. die Wirtschaftsprüfung. Innerhalb dieser sind viele Prozesse unstrukturiert und kommen deshalb nicht ohne menschliche Interaktion aus (Zhang et al. 2021, S. 5, 7 f.).

Im Gegensatz zu Attended RPA bezeichnet Unattended RPA einen RPA-Typen, bei dem die Software Roboter statt auf dem Desktop des Benutzers, auf einem Server bzw. auf einer virtuellen Maschine im Hintergrund ausgeführt werden. Sie können unabhängig von Menschen arbeiten und benötigen meistens keine direkte Interaktion mit diesen. Durch ihre Unabhängigkeit lassen sie sich auf Basis von einer vordefinierten Uhrzeit oder eines festgelegten Triggers, wie der Erhalt einer E-Mail, automatisch triggern. Unattended RPA-Roboter werden mithilfe eines Orchestrators gesteuert und überwacht, eine Schlüsselkomponente von RPA-Systemen. Beleuchtet man die Anwendungsfälle für Unattended RPA näher, so eignet sich dieser RPA-Typ besonders gut für regelbasierte Anwendungsfälle. So lassen sich Anwendungsfälle lassen sich im Rechnungswesen & Controlling Bereich finden, wenn z.B. per E-Mail erhaltene Rechnungen automatisch nach gewissen Regeln verbucht werden sollen (Zhang et al. 2021, S. 5, 7 f.; Langmann und Turi 2021, S. 6 f.; Axmann und Harmoko 2020, S. 559;

Choi et al. 2021, S. 3 f.)
Bei Hybridem RPA handelt es sich um eine Kombination von Attended und Unattended RPA. Es eignet sich vorwiegend für komplexe Prozesse, bei denen ein Teil vollautomatisiert ohne menschliche Interaktion abläuft, während der andere Teil des Prozesses auf die Interaktion mit einem Menschen angewiesen ist (Axmann und Harmoko 2020, S. 559 f.).
Generell besteht ein RPA-System i.d.R. aus drei Hauptkomponenten, dem RPA Studio, dem RPA Orchestrator und den RPA-Robotern. Der Aufbau eines solchen Systems entnommen werden. Das RPA Studio repräsentiert die Entwicklungsumgebung von RPA. Dort lassen sich die Prozesse, welche in Form eines Bots ausgeführt werden sollen, modellieren/ entwickeln und konfigurieren werden. Ist ein Bot fertig entwickelt, wird er dem Orchestrator übergeben. Der Orchestrator dient als zentrale Steuerungseinheit für die Verwaltung der Bots. Konkret ist er für die Planung, die Ausführung und das Monitoring der Bots zuständig. Der Orchestrator bietet in der Regel auch eine Schnittstelle, über die Anwendungen von Drittanbietern die RPA-Roboter nutzen können. Die Roboter führen dann die ihnen zugewiesenen Aufgaben aus (Choi et al. 2021, S. 4).

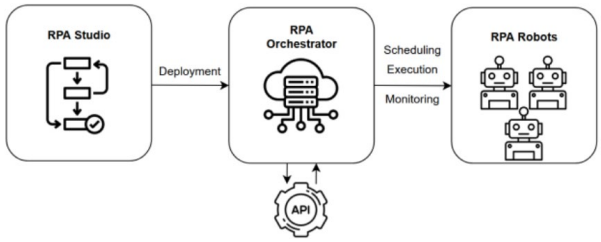


Abbildung 2 RPA-System Aufbau Quelle: Eigene Darstellung in Anlehnung an Choi et al. 2021, S. 4

Einsatzbereiche
RPA findet in einer Vielzahl von Bereichen Verwendung. Generell ist der Einsatz von RPA besonders geeignet für die Automatisierung von Geschäftsprozessen, die sich durch Regelmäßigkeit und Routinetätigkeiten auszeichnen. Weiter ist RPA geeignet für Prozesse, die strukturierte Daten verarbeiten, durch ein hohes Volumen geprägt sind und eine Interaktion mit mehreren IT-Systemen über die Oberfläche erfordern. RPA ist somit besonders für Prozesse geeignet, die keine Kreativität oder Interpretation erfordern (Da Costa et al. 2022, S. 8; Aguirre und Rodriguez 2017, S. 65 f., 70).
RPA findet in vielen unterschiedlichen Branchen Anwendung. Besonders häufig wird RPA in Bereichen wie IT, Personalwesen/HR, Versicherung, Buchhaltung und Finanzen, Einzelhandel sowie in der Wirtschaftsprüfung eingesetzt (Santos et al. 2020, S. 406; Kokina und Blanchette 2019, S. 1; Moffitt et al. 2018, S. 1, 9; Zhang et al. 2021, S. 2; Madakam et al. 2019, S. 1, 13). Die Aufgaben, die RPA dabei ausführt, sind oft das systematische Erfassen und Überprüfen von Daten, das Bearbeiten und Umstrukturieren von Dateien, das Anpassen von Formaten sowie das Synchronisieren und Abgleichen von Daten über mehrere Plattformen hinweg (Alberth und Mattern 2017, S. 58). Die aufgeführten Einsatzbereiche zeigen, dass RPA bereits eine tragende Rolle in der Optimierung und Automatisierung von Geschäftsprozessen einnimmt. Die Weiterentwicklung der RPA-Funktionalität erfolgt laufend, was unter anderem durch die Innovationen und Entwicklungen innerhalb des Informationstechnologiesektors vorangetrieben wird. Als Aktuelle Trends lassen sich folgende Trends ermitteln: Artificial Intelligence (AI) bzw. künstliche Intelligenz (KI), vermag die Effizienz zu steigern, indem die Integration von KI-Technologien RPA autonomer und dynamischer ausgestaltet. RPA kann hierdurch selbstständiger auf unterschiedliche Situationen reagieren und Abläufe kombinieren. Ein Beispiel hierfür ist die Verbesserung der Interaktion mit menschlichen Benutzern, indem empfangene Nachrichten korrekt interpretiert und automatisch beantwortet werden (Hanussek 2019). Weiter kann KI eine Verarbeitung von unstrukturierten Daten die Nutzung von Spracherkennung, die Verwendung von maschinellem Lernen und neuronalen Netzen im RPA Kontext ermöglichen. Neuronale Netze sind in der Lage, komplexe Muster in Daten zu erkennen, wodurch sich anspruchsvollere Aufgaben automatisieren lassen (UiPath o. D.; Köhler-Schute 2020, S. 29). Ein weiterer aktueller Trend ist die Verbindung von Process Mining mit RPA. Die Verwendung von Process Mining Software, wie z. B. Celonis, kann es Unternehmen ermöglichen, Prozesse leichter und effizienter zu identifizieren, die für eine Automatisierung im Kontext von RPA geeignet sind (Choi et al. 2022, S. 39604, 39611; Celonis o. D.). Ein weiterer aktueller Trend ist die Untersuchung des Einsatzes von Blockchain Technologien, z.B. bei Kryptowährungen wie Bitcoin oder Ethereum. Die Verbindung zu RPA ermöglicht es, Sicherheits- und Audit-Herausforderungen zu adressieren (Al-Slais und Ali 2023).

Vorteile und Herausforderungen
RPA hat sich als eine bedeutende Technologie innerhalb der digitalen Transformation etabliert. Jedoch besitzt RPA neben zahlreichen Vorteilen auch einige Herausforderungen, die Unternehmen bewältigen müssen, um bestmöglich von RPA zu profitieren.
Einer der wichtigsten Vorteile ist die erhebliche Effizienzsteigerung, die Unternehmen durch die Nutzung von RPA erreichen können. In einer Fallstudie wurde analysiert, wie effizient eine Gruppe unter Verwendung von RPA im Vergleich zu einer Gruppe ohne den Einsatz von RPA arbeitet. In dem Team, welche RPA-Software verwendet hat, konnten 21 % mehr Fälle bearbeitet werden als in der Vergleichsgruppe. Entsprechend eröffnet dies Wertsteigerungspotenziale, indem Mitarbieter die durch Automatisierung gewonnene Zeit für wichtigere und komplexere Aufgaben nutzen können. Im Rahmen der genannten Fallstudie ist allerdings zu ergänzen, dass die Gruppe mit RPA nicht signifikant schneller waren

als die ohne RPA (Aguirre und Rodriguez 2017, S. 68-70; Shidaganti et al. 2021, S. 1). Ein weiterer Vorteil von RPA für Unternehmen sind die damit verbundenen Kosteneinsparungen, welche unter anderem durch die Automatisierung der Geschäftsprozesse entstehen. Diese Kosteneinsparungen ergeben sich zum einen daraus, dass man Mitarbeiter gezielter und effizienter einsetzen kann aufgrund der Automatisierung von Routinetätigkeiten. Zusätzliche Einsparungen ergeben sich dadurch, dass RPA die Fehlerquote bei Aufgaben wie der Dateneingabe signifikant reduzieren kann. Hierdurch sinken die Fehlerbehebungskosten und zeitgleich steigt die Qualität der auszuführenden Arbeit (Ivančić et al. 2019, S. 280, 282, 287 ,290; Kirchmer 2017, S. 2 f.; Asatiani und Penttinen Esko 2016, S. 4). Zusätzlich kann ein Roboter im Gegensatz zu einem Mitarbeiter rund um die Uhr (24/7) arbeiten. Eine Studie aus dem Jahr 2016 zeigt auf, dass die Kosten für einen RPA-Roboter nur ein Drittel bis ein Fünftel der Kosten eines Vollzeitmitarbeiters betragen können (Kroll et al. 2016, S. 12; Asatiani und Penttinen Esko 2016, S. 4). Konkret können Unternehmen mit einer durchschnittlichen Kosteneinsparung von ca. 25 % rechnen. Der Break-Even-Point von RPA wird dabei meistens schon im ersten Jahr nach der Einführung erreicht, mit einem potenziellen Return on Investment (ROI) von 30 bis 200 % (Koch und Wildner 2020, S. 214; Lhuer 2016).

Ein weiterer Vorteil von RPA ist die Erhöhung der Compliance. Dies lässt darauf schließen, dass es für Unternehmen durch den Einsatz von RPA relativ einfach ist, sich an vorgegebene Regeln im Unternehmen zu halten und so ihre Gesamt-Compliance zu verbessern (Ivančić et al. 2019, S. 282). Ein weiterer Vorteil von RPA besteht darin, dass generell keine fortgeschrittenen Programmierkenntnisse benötigt werden, um einfache RPA-Roboter zu erstellen. Hinzu kommt, dass RPA von den Unternehmen ohne größere Anpassungen in die aktuelle IT-Landschaft integriert werden kann, da RPA nur auf dieser aufsetzt (Langmann und Turi 2021, S. 1, 8, 11).

Trotz der aufgezeigten Vorteile von RPA in Bezug auf Effizienzsteigerung und Kostenoptimierung sind auch spezifische Herausforderungen vorhanden. So eignet sich RPA nicht für alle Prozesse. RPA-Roboter sind nicht in der Lage, eigenständig Entscheidungen zu treffen. Aus diesem Grund stellen komplexe, unstrukturierte und stark variierende Prozesse eine Herausforderung für RPA-Roboter dar. Daraus kann abgeleitet werden, dass eine sorgfältige Analyse, Auswahl und Dokumentation der zu automatisierenden Prozesse innerhalb eines Unternehmens zu Beginn essenziell ist (Choi et al. 2021, S. 2 f.; Wanner et al. 2019, S. 2, 5; Langmann und Turi 2021, S. 14-16; Brettschneider 2020, S. 1103). Des Weiteren kann die Instandhaltung der RPA-Roboter für Unternehmen eine signifikante Herausforderung darstellen. RPA-Prozesse erfordern eine kontinuierliche Wartung, da selbst geringfügige Änderungen in den mit dem Prozess verbundenen Systemen und insbesondere in den Oberflächen, oft Anpassungen an den RPA-Robotern notwendig machen. Prozesse, die häufige Anpassungen benötigen, sind besonders fehleranfällig, was zu erhöhtem Wartungs- und Kostenaufwand führen kann (Langmann und Turi 2021, S. 14; Santos et al. 2020, S. 413). Je mehr Roboter im Einsat sind, desto höher wird auch der gesamte Wartungsaufwand innerhalb der RPA-Umgebung für ein Unternehmen (Brettschneider 2020, S. 1107). Durch die Automatisierung von Aufgaben, diezuvorvon menschlichen Mitarbeitern ausgeführt wurden, entfällt zukünftig ein Teil dieser Tätigkeiten aufgrund des Einsatzes von RPA. Daraus resultiert eine Herausforderung im Sinne der Workforce Resilience, d.h. des Widerstands der Mitarbeiter gegenüber dieser Technologie. Es besteht die Gefahr, dass diese befürchten, durch RPA ersetzt zu werden, was zu erhöhten Ängsten hinsichtlich Entlassungen führt. Unternehmen müssen rechtzeitig die Mitarbeiter in den Prozess der RPA-Einführung einbeziehen und ein geeignetes Change Management betreiben. Beispielsweise sollten geeignete Lösungswege und Fortbildungsmöglichkeiten aufgezeigt werden, um die RPA Einführung nicht zu gefährden (Syed et al. 2020, S. 8; Brettschneider 2020, S. 1104 f.; Santos et al. 2020, S. 414; Köhler-Schute 2020, S. 22). Weiterhin lässt sich das Management der Skalierbarkeit als potenzielles Problem identifizieren. So kann es innerhalb von RPA zu Schwierigkeiten kommen, wenn man versucht, eine einzelne Anwendung unternehmensweit auszurollen. Daher sollte bereits zu Beginn der Einführung ein Konzept bezüglich der Infrastruktur erarbeitet werden, um am Ende die gewünschte Skalierbarkeit und Stabilität der RPA-Umgebung erreichen zu können (Syed et al. 2020, S. 12; Langmann und Turi 2021, S. 61).

MICROSERVICES

Definition und Grundlagen

Viele Unternehmen stehen vor der Herausforderung, dass ihre IT-Landschaften nicht mehr zeitgemäß sind. Dies liegt daran, dass sie zu einer Zeit implementiert wurden, als Aspekte wie Modularisierung noch nicht im Fokus standen. Dies erfordert eine Veränderungsbereitschaft, um wettbewerbsfähig bleiben zu können (Dowalil 2018, S. 17 f.; Habibullah et al. 2019, S. 1 f.). Dieser Wandel spiegelt sich insbesondere in der Entwicklung der Software-Architektur wider. In der Vergangenheit waren monolithische Architekturen, bei denen die gesamte Software als ein einziger großer Codeblock umgesetzt wurde, weitverbreitet. Dieser umfangreiche Block erfüllte funktionale und nicht-funktionale Anforderungen, was als Resultat eine enge Verknüpfung zwischen den einzelnen Bestandteilen mit sich brachte. Diese Herangehensweise wird jedoch, besonders bei zunehmender Komplexität, unwirtschaftlich, kostenintensiv und zeitaufwändig. Der Grund dafür liegt in erhöhten Ressourcenaufwänden und verlängerten Entwicklungszeiten. Entsprechend wurden modulare Architekturen entwickelt, um größere Flexibilität und Skalierbarkeit zu erzielen (Yousif 2016, S. 4; Kalske et al. 2018, S. 32 f.). In diesem Zusammenhang spielte die "Service-Oriented Architecture" (SOA) eine bedeutende Rolle. SOA ist eine Architekturform, bei der die Systemlandschaft in einzelne, unabhängige Services zerlegt wird.

Jeder Service ist dabei genau für eine Geschäftsaufgabe zuständig. Diese Services werden innerhalb eines Netzwerks verteilt und kommunizieren über definierte Schnittstellen, wodurch eine lose Kopplung zwischen ihnen entsteht. Unter einer losen Kopplung wird die Minimierung der Verbindungen zwischen einzelnen Softwarekomponenten verstanden. Dies impliziert, dass zwischen den verschiedenen Bestandteilen einer Software möglichst wenige Abhängigkeiten bestehen sollten, damit z. B. in Fehlerfällen nicht das ganze System betroffen ist. Aufbauend auf den Konzepten von SOA haben sich Microservices als eine modulare Unterart bzw. als ein spezifischer Typ einer "Service-Oriented Architecture" entwickelt. (Dowalil 2018, S. 25-29, 121 f., 196; Newman 2020, S. 1). Heutzutage gewinnen Microservices zunehmend an Popularität und werden bereitsvon einer Vielzahl von Unternehmen implementiert (Taibi et al. 2017, S. 23). Wird die Definition eines Microservices betrachtet, so ist festzustellen, dass bisher keine einheitliche Definition existiert. Auf Basis der Definitionen von Fowler und Lewis sowie der Definition von Newman lässt sich jedoch eine allgemeine Charakterisierung ableiten: Microservices sind kleine, unabhängige und lose gekoppelte Einheiten, die jeweils eine spezifische Funktion oder Aufgabe erfüllen. Sie kommunizieren über geeignete Schnittstellen miteinander und sind stark gekapselt, was ihre unabhängige Veröffentlichung und Wartung ermöglicht. Wenn in einem System mehrere solcher Microservices zusammenarbeiten, spricht man von einer Microservice-Architektur. Betrachtet man die Größe eines Microservices näher, so könnte der Begriff „Micro“ eine geringe Größe suggerieren. Allerdings gibt es keine klar definierte Größe für einen Microservice. Die tatsächliche Größe hängt stark vom jeweiligen Kontext ab, weshalb ein einzelner Service nicht zwingend klein sein muss (Fowler und Lewis 2014; Newman 2020, S. 1, 9 f.; Nadareishvili et al. 2016, S. 65 f.) Abbildung 3 zeigt den beispielhaften Aufbau und die Funktionsweise einer solchen.

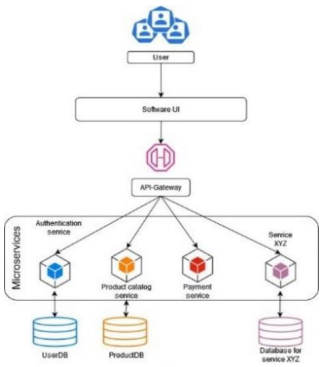


Abbildung 3 Microservice-Architektur Quelle: Eigene Darstellung in Anlehnung an Microsoft Learn o. D., 2023; Lal Sahni 2023

In einer Microservice-Architektur greifen Benutzer über eine Benutzeroberfläche auf die Software zu, die im Hintergrund über ein API-Gateway mit den benötigten Microservices kommuniziert. Das API-Gateway leitet Anfragen weiter und koordiniert die Antworten. Ein Beispiel hierfür ist der Authentifizierungsservice, der den Benutzerlogin als eigenständigen Microservice abwickelt. Microservices sind unabhängig voneinander entwickelbar und skalierbar. Skalierbarkeit erfolgt entweder vertikal durch die Erweiterung der Ressourcen einer Maschine oder horizontal durch die Verteilung der Arbeitslast auf mehrere Systeme (RedHat 2019; AWS o. D.; Blinowski et al. 2022, S. 20360).

Architekturprinzipien

Für die effektive Integration von Microservices in ein System sind bestimmte Architekturprinzipien unerlässlich, um ihr volles Potenzial zu entfalten. Laut der IEEE/ISO/IEC 42010-2022 Norm beschreibt eine Architektur das zentrale Konzept und die charakteristischen Merkmale einer Einheit in einem definierten Umfeld. Sie enthält alle notwendigen Informationen für Implementierung und Weiterentwicklung. Gartner ergänzt diese Definition, indem die verwendete Hardware, Software und Kommunikationsmechanismen als entscheidende Bestandteile der Architektur hervorgehoben werden (IEEE Computer Society/Software & Systems Engineering Standards 2022; Gartner o. D.a).

Das erste zentrale Architekturprinzip von Microservices ist die Modularität. Dabei wird ein System in klar abgegrenzte Module aufgeteilt, die durch ihre externen Eigenschaften, insbesondere ihre Schnittstellen, definiert sind. Diese Schnittstellen ermöglichen die Interaktion zwischen den Modulen. Ein wichtiger Aspekt der Modularität ist die Austauschbarkeit von Modulen. Ein Modul kann durch ein anderes ersetzt werden, solange es dieselben Eigenschaften aufweist. Jedes Modul sollte zudem leicht weiterentwickelbar sein und über eine eigene Dokumentation verfügen (Dowalil 2018, S. 2 f.). Ein weiteres wichtiges Architekturprinzip ist die Kontextgrenze, auch „Bounded Context“ genannt, die aus dem Domain-Driven Design stammt. Dieser Softwareentwicklungsansatz zielt darauf ab, Software entlang der Prozesse und Regeln einer bestimmten Domäne zu entwickeln. Die Kontextgrenze definiert klare Grenzen für ein Modul oder eine Microservice-Komponente und stellt sicher, dass das Modell innerhalb dieser Grenzen einheitlich und konsistent bleibt. Dabei geht es nicht nur um die Bereitstellung einer spezifischen Funktionalität, sondern auch darum, die interne Komplexität zu verbergen, sodass externe Systeme keinen Zugriff auf interne Details erhalten (Newman 2021, S. 52 f., 58 f., 2020, S. 31; Dowalil 2018, S. 64 f.; Fowler 2020). Um die Modularität in einer Architektur effektiv umzusetzen, spielen ebenfalls Software-Design-Prinzipien wie Separation of Concerns und das Single Responsibility Principle eine zentrale Rolle. Das Prinzip der Separation of Concerns besagt, dass jede Funktion eines Systems in einem eigenständigen Baustein realisiert werden sollte. Im Kontext von Microservices bedeutet dies, dass jede Funktionalität als eigenständiger Service abgebildet wird. Das Single Responsibility Principle ergänzt dieses Konzept, indem es sicherstellt, dass jeder Microservice nur eine spezifische Verantwortlichkeit hat. Dadurch existiert für jeden Service nur einen Grund für Änderungen, was die Wartbarkeit und Weiterentwicklung vereinfacht (Dowalil 2018, S. 31

ff.; Hu 2023, S. 107). Das zweite wichtige Architekturprinzip ist die lose Kopplung. Grundsätzlich beschreibt die Kopplung die Abhängigkeiten zwischen den Bausteinen einer Architektur. Im Kontext von Microservices bedeutet lose Kopplung, dass die einzelnen Services nur minimale Informationen übereinander besitzen und weitestgehend unabhängig agieren. Konkret heißt das, dass Änderungen an einem Service keine oder nur minimale Auswirkungen auf andere Services haben sollten. Eine stabile Struktur zeichnet sich dadurch aus, dass sie starke Kohäsion und schwache Kopplung aufweist. Kohäsion beschreibt das Maß, in dem die Komponenten innerhalb eines Moduls eng miteinander verbunden und auf eine gemeinsame Funktionalität ausgerichtet sind. Eine hohe Kohäsion fördert die Wartbarkeit und Weiterentwicklung der Module, da die Funktionen eines Moduls klar definiert und aufeinander abgestimmt sind (Newman 2020, S. 17; Dowalil 2018, S. 25; Newman 2021, S. 38 f.; Farley o. D.). Neben der Modularität und der losen Kopplung ist die Autonomie der Services ein weiteres entscheidendes Prinzip. Die Autonomie lässt sich aus zwei Perspektiven betrachten. Zum einen sollten verschiedene Teams in der Lage sein, im Rahmen einer gemeinsamen Governance (Shared Governance) eigenständig Services zu entwickeln und bereitzustellen. Das bedeutet, dass diese Teams volle Verantwortung für die Entwicklung und Verwaltung ihrer Services tragen. Zum anderen sollten die Microservices selbst ebenfalls autonom sein. Dies erfordert, dass jeder Service so gestaltet ist, dass er unabhängig und in sich geschlossen funktioniert. Dadurch kann ein Microservice losgelöst von anderen Services entwickelt, veröffentlicht und betrieben werden, was die Flexibilität und Skalierbarkeit der gesamten Architektur erheblich steigert (Khan et al. 2021, S. 7). Ein weiteres zentrales Architekturprinzip von Microservices betrifft die Kommunikation zwischen den einzelnen Services. Microservices interagieren über Schnittstellen (APIs), die eine entscheidende Rolle im System spielen, da sie die interne Kommunikation zwischen den Services ermöglichen. Um eine lose Kopplung zu gewährleisten, sollten diese Schnittstellen lediglich die Informationen bereitstellen, die für die Interaktion mit anderen Services notwendig sind. Dies reduziert Abhängigkeiten und fördert die Unabhängigkeit der einzelnen Microservices (Dowalil 2018, S. 123). Im Kontext von Microservices hat sich hier das Designprinzip „smart endpoints and dumb pipes“ durchgesetzt. Jeder Microservice fungiert dabei als eine Art Filter. Er empfängt Anfragen, verarbeitet sie mit der entsprechenden Logik und liefert das entsprechende Ergebnis zurück. Die Kommunikation zwischen den Services wird dabei möglichst einfach gehalten, ohne komplexes Anfragenrouting oder Datentransformationen. Zur Kommunikation werden hauptsächlich zwei Protokolle verwendet: HTTP für synchrone Kommunikation, bei der eine Anfrage und eine direkte Antwort erfolgen, und Lightweight Messaging für asynchrone Kommunikation über einen Nachrichtenbus (Alpers et al. 2015, S. 73; Fowler und Lewis 2014; Dowalil 2018, S. 74 f.; Montemagno et al. 2022). Obwohl das Designprinzip „smart endpoints and dumb pipes“ die Komplexität reduziert und eine einfache Kommunikation zwischen den Services fördert, bleibt die Möglichkeit von Fehlern bestehen. Diese können durch menschliches Versagen oder technische Störungen verursacht werden. Daher ist Resilienz ein zentraler Aspekt beim Entwurf einer Microservice-Architektur. Resilienz beschreibt die Fähigkeit eines Systems, trotz auftretender Fehler weiter zu funktionieren und sich schnell zu erholen. Das Ziel ist es, dass bei einem Fehler nicht das gesamte System ausfällt, sondern nur die betroffenen Bereiche beeinträchtigt werden, während der Rest des Systems weiterhin funktionsfähig bleibt. Zur Unterstützung der Resilienz können Mechanismen wie Timeouts implementiert werden. Dabei wird festgelegt, dass eine Serviceanfrage innerhalb einer bestimmten Zeit beantwortet werden muss. Bleibt diese aus, wird der Service als ausgefallen betrachtet (Indrasiri und Siriwardena 2018, S. 42; Wolff 2018, S. 207 f.). Damit ein reibungsloser Betrieb in einer Microservice-Architektur gewährleistet werden kann, sind Monitoring- und Logging-Funktionalitäten unerlässlich. Logging ermöglicht es, Ereignisse in den einzelnen Services nachzuvollziehen, was nicht nur für die Erstellung von Statistiken, sondern vor allem für die effiziente Fehlersuche von zentraler Bedeutung ist. Log-Dateien sind in der Regel so strukturiert, dass sie von Menschen leicht interpretiert werden können. Durch Monitoring können wichtige Metriken wie die Antwortzeiten der Services und die Anzahl fehlgeschlagener Anfragen kontinuierlich überwacht werden. Diese Informationen sind entscheidend, um frühzeitig Probleme zu identifizieren und die Systemstabilität sicherzustellen (Wolff 2018, S. 244 f.; Indrasiri und Siriwardena 2018, S. 48 f., 373; Khan et al. 2021, S. 8). Aufgrund der erhöhten Netzwerkkommunikation in einer Microservice-Architektur ist es essenziell, dass nur autorisierte Parteien Zugriff auf die Services haben. Daher sollten Microservices nur die minimal benötigten Rechte besitzen, insbesondere bei Datenbankzugriffen und sensiblen Ressourcen. Ein effizientes Identity- und Access Management (IAM) ist unerlässlich, um sicherzustellen, dass ausschließlich autorisierte Nutzer und Services auf die Microservices zugreifen können. Zudem müssen Zugangsdaten wie E-Mails und Passwörter sicher gespeichert werden und dürfen nicht im Klartext vorliegen, um Sicherheitsrisiken zu vermeiden (Newman 2021, S. 29, 345-347, 354-456).

Vorteile und Herausforderungen

Microservices bieten gegenüber monolithischen Architekturen zahlreiche Vorteile, jedoch besitzen sie auch Herausforderungen. Ein wesentlicher Vorteil besteht in der Unterstützung agiler Arbeitsweisen, die es Unternehmen ermöglicht, schneller auf veränderte Geschäftsanforderungen zu reagieren. Innerhalb einer Microservice-Architektur lassen sich neue Services leichter entwickeln, bereitstellen und testen, was eine höhere Flexibilität sowie schnellere Iterationen ermöglicht. Falls ein neu entwickelter Service nicht den gewünschten Anforderungen entspricht, kann dieser mit geringem Aufwand deaktiviert und durch eine alternative Implementierung ersetzt werden. Diese Agilität verkürzt die Entwick-

lungszeit von Microservices erheblich, was es den Entwicklerteams ermöglicht, flexibler und schneller auf Marktveränderungen zu reagieren. Dadurch können Unternehmen ihre Innovationszyklen beschleunigen und Wettbewerbsvorteile erzielen (Indrasiri und Siriwardena 2018, S. 14; Nadareishvili et al. 2016, S. 14-16; Khan et al. 2021, S. 11). Ein weiterer Vorteil von Microservices ist die verbesserte Wartbarkeit. Dank der modularen und unabhängigen Struktur lassen sich Services leichter warten. Da die Logik und Datenhaltung innerhalb der Services erfolgt, sind keine externen Abhängigkeiten notwendig. Updates betreffen somit nur den jeweiligen Service, während der Rest des Systems unverändert bleibt. Dies fördert nicht nur die Wartbarkeit, sondern auch die langfristige Nachhaltigkeit der Softwareentwicklung, da veraltete Systemstrukturen vermieden und eine zukunftsfähige IT-Infrastruktur geschaffen wird (Khan et al. 2021, S. 11; Eyerman und Hur 2022, S. 1; Wolff 2018, S. 60-62). Neben der Wartbarkeit ermöglichen Microservices Unternehmen, neue Features und Bugfixes schneller zu veröffentlichen. Die Unabhängigkeit der Entwicklungsteams trägt dazu bei, da sie ohne Abhängigkeit von anderen Teams effizienter arbeiten können. Zudem erlaubt die Microservice-Architektur den Entwicklern, moderne Technologien zu nutzen, ohne an veraltete Entscheidungen gebunden zu sein. Ein weiterer Vorteil ist die Skalierbarkeit: Microservices können individuell skaliert werden, ohne das gesamte System anpassen zu müssen. Dies ermöglicht eine effiziente Nutzung von Ressourcen, indem Services bei steigenden Anfragen hoch- und bei sinkenden Anfragen wieder herunterskaliert werden, ohne die Performance zu beeinträchtigen (Khan et al. 2021, S. 11, 12; Wolff 2018, S. 64 f., 66 f.; GitLab 2022). Trotz der Vorteile wie gesteigerte Agilität, verbesserte Wartbarkeit, verkürzte Time-to- Market und erhöhter Skalierbarkeit müssen auch die Herausforderungen von Microservices berücksichtigt werden. Eine zentrale Herausforderung ist die erhöhte Komplexität, die durch die Vielzahl autonomer und lose gekoppelter Komponenten entsteht. Insbesondere die Inter-Service-Kommunikation erfordert effiziente und zuverlässige Kommunikationswege, die oft komplexer sind als die Entwicklung der Services selbst. Auch das Daten- und Transaktionsmanagement wird durch die verteilte Logik und Datenhaltung anspruchsvoller. Zusätzlich ist bereits die Definition und Erstellung der Services eine Herausforderung. Um eine effektive Modularisierung sicherzustellen, müssen Unternehmen klar identifizieren, welche Funktionalitäten als eigenständige Module umgesetzt werden sollten. Falsch definierte Servicegrenzen können zu erhöhtem Datenaustausch über das Netzwerk führen, was wiederum die Kopplung zwischen den Services verstärkt und der Grundidee der losen Kopplung widerspricht (Indrasiri und Siriwardena 2018, S. 15 f.; Jamshidi et al. 2018, S. 31).

Auch das Monitoring der Services stellt eine Herausforderung dar. Durch die Verteilung der Microservices kann es schwierig sein, den Überblick über die Beziehungen zwischen den einzelnen Services zu behalten. Die autonome Struktur der Services, die zudem in unterschiedlichen Technologien implementiert sein können, macht ein umfassendes Logging- und Monitoringkonzept unerlässlich. Dies wird zusätzlich dadurch erschwert, dass Microservices oft von unterschiedlichen Teams entwickelt werden. Eine weitere Herausforderung sind die Kosten bei der Einführung einer Microservice-Architektur, die vor allem in der Anfangsphase hoch ausfallen können. Diese resultieren unter anderem aus der Notwendigkeit, die bestehende Infrastruktur zu erweitern, etwa durch den Ausbau des Netzwerks, zusätzlichen Speicherplatz oder die Implementierung zusätzlicher Software. Zudem müssen während der Migration die monolithischen Systeme parallel zu den neuen Microservices betrieben werden, was den Aufwand weiter erhöht. Ein weiterer Faktor sind die verzögerten Entwicklungsprozesse zu Beginn der Umstellung, da sich Entwickler zunächst an die neuen Strukturen und Arbeitsabläufe gewöhnen müssen. Abhängig vom Erfahrungsgrad der Entwickler können zusätzliche Kosten für die Einstellung von Fachkräften mit Microservice-Kenntnissen entstehen. Trotz dieser Anfangskosten sollten Microservices jedoch als strategische Investition betrachtet werden, die langfristig Effizienzsteigerungen und Flexibilität fördern (Niedermaier et al. 2019, S. 42-44; Khan et al. 2021, S. 14; Baškarada et al. 2020, S. 6; Newman 2021, S. 26-28; Singleton 2016, S. 17).

USECASE

Vorgehensweise Experteninterviews

Zur Erfassung der Anforderungen an die zu entwickelnde Microservice-Architektur wurden leitfadenbasierte Experteninterviews durchgeführt. Ziel dieser Interviews war es, relevante Anforderungen zu identifizieren und wertvolle Einblicke aus der Praxis zu gewinnen, die aus anderen Quellen nur schwer zu ermitteln wären. Der Interviewleitfaden ist in drei Bereiche unterteilt:

1. Allgemeine Informationen der Experten: Zunächst wurden Fragen gestellt, um den Hintergrund der Experten zu erfassen und ihre Erfahrungen im jeweiligen Unternehmenskontext besser einzuordnen.
2. Kenntnisse über Microservices und Robotic Process Automation (RPA): In diesem Abschnitt wurden die Experten zu ihrer Einschätzung hinsichtlich verschiedener relevanter Themenbereiche innerhalb einer Microservice-Architektur befragt. Dabei sollten sie die Ziele, die mit einer solchen Architektur verfolgt werden können, nach Wertbeitrag und Realisierbarkeit priorisieren, um die Anforderungen praxisnah und umsetzbar zu gestalten.
3. Spezifische Anforderungen an die Architektur: Abschließend wurde ein tieferer Einblick in die spezifischen Erwartungen und Anforderungen der Experten gewonnen. Hierbei lag der Fokus auf den wichtigsten Mehrwerten für die zukünftige Architektur sowie auf potenziellen Methoden zur Realisierung und Katalogisierung der Microservices.

Die Interviews wurden per Videokonferenz durchgeführt, um eine strukturierte Erhebung der Anforderungen

sicherzustellen.

Ergebnisse der Experteninterviews

Die befragten Experten setzten sich aus Beratern und Entwicklern zusammen, was zu unterschiedlichen Ansichten führte. Die Interviews zeigten, dass das Verständnis von Microservices und RPA variiert. Einige Experten sehen RPA als Werkzeug zur Automatisierung von Frontend-Prozessen, während andere eine breitere Definition verwenden, die auch die Automatisierung von Prozessen in Systemen ohne native Automatisierungsfunktionen umfasst. Beim Thema Microservices reichten die Auffassungen von einer einfachen modularen Architektur bis hin zur Weiterentwicklung der serviceorientierten Architektur (SOA), die komplexere Systemarchitekturen ermöglicht. Ein zentrales Ergebnis der Interviews ist die Identifikation der Hauptziele bei der Implementierung einer Microservice-Architektur im RPA-Umfeld. Dabei wurden Skalierbarkeit, Flexibilität, Wartbarkeit und Performance als zentrale Prioritäten genannt. Außerdem bewerteten die Experten die potenziellen Mehrwerte nach Wertbeitrag und Realisierbarkeit, was eine systematische Rangordnung ermöglichte. Besonders hoch evaluiert wurden die Mehrwerte „Wiederverwendbarkeit“, „Wartbarkeit“, „Skalierbarkeit“, „Ersetzbarkeit“ und „Flexibilität“. Diese spielen eine entscheidende Rolle bei der Architektur entwicklung (siehe Tabelle 1). Ein weiterer Schwerpunkt der Untersuchung lag auf der Analyse, wie die einzelnen Expertengruppen, insbesondere Entwickler und Berater, die Mehrwerte unterschiedlich bewerten. Hier zeigte sich eine weitgehende Übereinstimmung der Top-5-Mehrwerte beider Gruppen. Jedoch wurden Un terschiede hinsichtlich der Gewichtung einzelner Mehrwerte beobachtet: Während Entwickler vermehrt die „Stabilität“ der Systeme in den Vordergrund stellten, legten Berater einen stärkeren Fokus auf die „Flexibilität“ der Architektur.

Tabelle 1 Top-Mehrwerte Quelle: Eigene Darstellung

Reihenfolge	Top-Mehrwerte	Gesamtpunktzahl durch Experten
1	Wiederverwendbarkeit	675
2	Wartbarkeit	544
3	Skalierbarkeit	501
4	Ersetzbarkeit	484
5	Flexibilität	480
6	Testbarkeit	428
7	Performance	403
8	Stabilität	396
9	Fehlerbehebungsmöglichkeiten	351
10	Integrationsfähigkeit	224
11	Time-to-Market	146
12	Compliance und Governance	128
13	Administrierbarkeit	104
14	Sicherheit	92

Die detaillierte Analyse der Mehrwertbewertungen verdeutlicht, dass insbesondere die „Wiederverwendbarkeit“ und „Wartbarkeit“ von allen Experten als sehr relevant eingestuft wurden. Dies lässt sich darauf zurückführen, dass diese Eigenschaften eine langfristige Kostenersparnis und höhere Effizienz bei der Wartung und Erweiterung der Architektur ermöglichen. Die Unterschiede zwischen Entwicklern und Beratern in der Priorisierung anderer Merkmale, wie „Stabilität“ versus „Flexibilität“, lassen sich durch die jeweiligen Rollen und Verantwortlichkeiten der Experten erklären. Entwickler fokussieren sich in ihrer Arbeit häufig auf die technische Umsetzbarkeit und die Gewährleistung der Systemstabilität, während Berater strategische Faktoren wie Anpassungsfähigkeit und Skalierbarkeit im Kontext zukünftiger Anforderungen stärker gewichten.

Zusammenfassend lässt sich festhalten, dass die Experteninterviews wertvolle Einblicke in die unterschiedlichen Einschätzungen und Prioritäten hinsichtlich der Mehrwerte von Microservices und RPA liefern. Die Ergebnisse deuten darauf hin, dass trotz unterschiedlicher beruflicher Hintergründe zentrale Mehrwerte von allen Expertengruppen ähnlich hoch gewichtet werden. Dies legt nahe, dass bestimmte Eigenschaften, wie Wartbarkeit und Wiederverwendbarkeit, als universell bedeutend für die Implementierung von Microservice-Architekturen in Verbindung mit RPA gelten.

Anforderungen und Anpassung der Architekturprinzipien

Die abgeleiteten Anforderungen an eine Microservice-Architektur im RPA-Umfeld werden auf Basis eines kategorienbasierten Ansatzes abgeleitet. In der Kategorie „Entwicklung und Design“ stehen Modularität, Wiederverwendbarkeit, Flexibilität und lose Kopplung im Vordergrund, um die Anpassungsfähigkeit und Effizienz der Architektur zu maximieren. Im Bereich „Betrieb und Wartung“ werden die Wartbarkeit, einfache Verwaltung und umfassende Dokumentation hervorgehoben, um eine langfristig stabile und pflegeleichte Architektur sicherzustellen. Für die Kategorie „Performance und Qualität“ waren vor allem Skalierbarkeit, Stabilität und Robustheit von zentraler Bedeutung, da diese eine hohe Effizienz und Zuverlässigkeit auch unter variierenden Bedingungen garantierten. In der Kategorie „Governance und Compliance“ liegt der Fokus auf der Entwicklung von Richtlinien, einer Katalogisierung der Microservices sowie der Verantwortung für die Einhaltung von Standards und deren regelmäßiger Überprüfung.

Die zuvor beschriebenen Anforderungen werden durch angepasste Architekturprinzipien in der Microservice-Architektur für RPA abgebildet. Da RPA-Microservices noch einen neuen Ansatz darstellen, müssen die allgemeinen Prinzipien für Microservices an die spezifischen Gegebenheiten angepasst werden.

Die Modularität ist so anzupassen, dass die RPA-Microservices flexibel, leicht austauschbar und wiederverwendbar sind. In der Architektur wird dies dadurch umgesetzt, dass jedes Modul über wenige Parameter (wie Modulname und Inputparameter) aufgerufen werden kann. Durch diese Struktur können einzelne Module nahtlos durch andere mit denselben Eigenschaften ersetzt werden.

Das Prinzip des Bounded Contexts wird übernommen, um sicherzustellen, dass die Komplexität der Automatisierungsprozesse verborgen bleibt. In der Architektur wird dies durch die präzise Abgrenzung der RPA-Microservices erreicht, sodass jeder Service eine spezifische Aufgabe übernimmt, ohne Details seiner internen Funktionsweise nach außen preiszugeben. Dies verstärkt die Anwendung der Prinzipien Separation of Concerns

und Single Responsibility: Diese stellen sicher, dass jeder Microservice eine klar abgegrenzte Funktion erfüllt. In der Architektur bedeutet dies, dass RPA-Microservices als isolierte, unabhängig operierende Einheiten entworfen werden, was die Wartbarkeit und Integration erleichtert.
Die lose Kopplung wird dahingehend angepasst, dass RPA-Microservices möglichst unabhängig voneinander agieren. In der Architektur erfolgt die Umsetzung durch die Schaffung von isolierten Automatisierungsprozessen, die nur minimale Abhängigkeiten zu anderen Prozessen aufweisen. Dies ermöglicht es, Änderungen an einem RPA-Prozess vorzunehmen, ohne dass andere Prozesse oder der Gesamtprozessfluss beeinträchtigt werden, solange die definierten Schnittstellen (Input- und Outputparameter) unverändert bleiben. Trotz der engen Integration mit den zu automatisierenden Systemen bleibt die lose Kopplung durch die klare Trennung der Automatisierungsaufgaben bestehen.
Die Autonomie der Services wird ebenfalls angepasst. Jeder RPA-Service ist so konzipiert, dass er unabhängig von anderen Services betrieben und veröffentlicht werden kann. In der Architektur wird dies durch den Einsatz von einer Orchestrierungssoftware unterstützt, die die Koordination der RPA-Services ermöglicht. Dadurch können die RPA-Services flexibel in unterschiedliche Geschäftsprozesse integriert und wiederverwendet werden, was die Autonomie und Skalierbarkeit der Architektur erhöht. Die Prinzipien der Resilienz, Monitoring und Logging werden so angepasst, dass sie den spezifischen Anforderungen von RPA entsprechen. In der Architektur erfolgt die Abbildung durch die Nutzung von RPA-Plattformen und Orchestrierungsumgebungen, die Mechanismen zur Überwachung des Zustands und der Performance der RPA-Prozesse bieten. Diese Systeme ermöglichen es, Fehlerszenarien aufzuzeichnen und die Stabilität der Microservices sicherzustellen. Logging und Monitoring werden genutzt, um Aktivitäten nachzuvollziehen und für Compliance- und
Audit-Zwecke zu protokollieren. Beim Identity Management wird das Prinzip angepasst, um eine sichere Verwaltung von Zugriffsrechten in der Architektur zu gewährleisten. Die Umsetzung erfolgt durch die in RPA- und Orchestrierungsplattformen integrierten Lösungen, die sicherstellen, dass nur autorisierte Nutzer und Systeme auf bestimmte Ressourcen zugreifen können. Die sichere Speicherung von Zugangsdaten erfolgt dabei durch verschlüsselte Verfahren, um die Sicherheit der Architektur weiter zu erhöhen.

Microservice-Architekturentwurf
Aufbau und Komponenten
Die entwickelte Microservice-Architektur bildet die Grundlage für die Modularisierung von RPA-Prozessen und zielt darauf ab, eine flexible, skalierbare, modulare, ersetzbare, wiederverwendbare und wartbare Umgebung zu schaffen.
Die Architektur unterscheidet zwischen zwei Arten von Microservices:

- Wertgenerierende Microservices: Diese Services verarbeiten Daten nach festen Regeln und liefern ein messbares Ergebnis, z. B. das Verarbeiten einer Excel-Tabelle und das Bereitstellen eines neuen Outputs.
- Nicht-wertgenerierende Microservices: Diese Services unterstützen den Gesamtprozess, tragen jedoch nicht direkt zum Endergebnis bei, wie z. B. die Authentifizierung an Systemen.

Im RPA-Kontext ist diese Unterscheidung wichtig, da nach jedem abgeschlossenen Prozess eine erneute Authentifizierung erforderlich ist. Die Architektur sieht daher vor, nicht-wertgenerierende Services als standardisierte Module bereitzustellen, die zentralisiert angepasst werden können.
Wertgenerierende Microservices werden hingegen als eigenständige RPA-Prozesse umgesetzt. Jeder Prozess fungiert als eigenständiger Microservice, der spezifischen Input verarbeitet und daraufhin einen entsprechenden Output generiert. Diese Services werden über ein Orchestrierungstool genutzt, wobei die Gesamtprozessmodellierung in einem Business Process Automation Tool erfolgt. Die Kommunikation erfolgt dabei über zuvor vordefinierte Schnittstellen, die den Prozessoutput zurückgeben.

Katalogisierung
Die Anzahl der Microservices kann in Unternehmen schnell steigen, besonders bei dezentralen Organisationsstrukturen. Ein zentraler Katalog bietet hierbei eine hilfreiche Übersicht zu Funktionalitäten und Zuständigkeiten der Microservices. Der ausgearbeitete Katalog dient der Verwaltung und Weiterentwicklung und basiert auf den geführten Experteninterviews. Zu den zentralen Katalogelementen gehören grundlegende Informationen wie der Servicename, eine klare Beschreibung der Entwickler und Verantwortlichen. Diese Daten helfen, Zuständigkeiten zu klären und SLAs festzulegen. Technische Details wie Inputparameter, erwarteter Output und Abhängigkeiten ermöglichen eine klare Nutzungsübersicht. Wichtig ist es, dort die aktuelle Version sowie die letzte Änderung zu dokumentieren, inklusive einer Verlinkung zu weiterführenden Informationen (z.B. einer Wiki-Seite). Für die Umsetzung bietet sich eine Tabellenstruktur in einer Wiki-Umgebung wie Confluence an, um die Übersichtlichkeit zu wahren. Mit steigender Anzahl an Microservices wird die Nutzung eines Marktplatzes empfohlen. Ein Beispiel ist der UiPath-Marktplatz, bei dem Services nach Kategorien filterbar sind. Eine mögliche Umsetzung kann z.B. mit dem Open-Source-Tool „Backstage“ von Spotify erfolgen, das über einen zentralen Softwarekatalog Metadaten und Abhängigkeiten von Microservices verwaltet und visualisieren kann.

Governance und Maintenance
Governance definiert die Richtlinien und Standards, die für den Betrieb und die Verwaltung der Architektur notwendig sind. Wartung bezieht sich auf die kontinuierliche Aktualisierung und Optimierung der Microservices, um

deren Leistung und Zuverlässigkeit sicherzustellen. Es wird eine Shared Governance empfohlen, bei der Teams innerhalb eines festgelegten Rahmens selbstständig Entscheidungen treffen können und Verantwortung für ihre Microservices übernehmen. Dabei sind klare Standards, insbesondere in Bezug auf Sicherheit, Dokumentation und Zugriffsrichtlinien, von zentraler Bedeutung. Jeder Microservice sollte modular, skalierbar und nur mit dem notwendigen Funktionsumfang ausgestattet sein. Bereits bestehende Monitoring- und Logging-Systeme sollten genutzt und nach einer Testphase bewertet werden. Um die Einhaltung der Standards sicherzustellen, wird die Einrichtung eines Governance-Komitees vorgeschlagen, das als zentrale Anlaufstelle fungiert und die Implementierung der Richtlinien überwacht. Microservices sind regelmäßig zu überprüfen und zu optimieren, beispielsweise je nach Art des Services halbjährlich oder jährlich. Diese Überprüfungen sollten durch automatisierte Tests unterstützt werden, die regelmäßig erfolgen. Zusätzlich wird empfohlen, Schulungs- und Weiterbildungsprogramme für Mitarbeiter anzubieten, um ein tieferes Verständnis von Microservices, den zugrundeliegenden Technologien und Best Practices im gesamten Unternehmen zu fördern. Dies trägt nicht nur zur Verbesserung der fachlichen Kompetenz bei, sondern unterstützt auch die fortlaufende Optimierung der Architektur.

Integration

Die bestehende Beispiel-Infrastruktur nutzt UiPath als RPA-Tool und Flowable als BPA-Tool. Diese beiden Systeme spielen eine zentrale Rolle bei der Integration im Bebauungsplan.

Die Architektur unterscheidet zwischen wertgenerierenden und nicht-wertgenerierenden Microservices. Für wertgenerierende Microservices wird ein vierstufiges Modell verwendet, das aus den Schichten Workflow, Orchestrator, Automation Control Center und Automation Bot besteht. Die Workflow-Schicht stellt die Gesamtprozessmodellierung in Flowable dar, wo Prozesse nach dem BPMN 2.0-Standard modelliert werden. UiPath Microservices werden in den Workflow integriert und lassen sich über eine Service-Task ansteuern. Diese Service-Task kommuniziert über einen Konnektor mit dem UiPath Orchestrator und führt API-Aufrufe aus, um den entsprechenden RPA-Prozess zu starten. Der Konnektor informiert bei Fehlern und bietet eine Restart-Option, um die Robustheit der Prozesse zu gewährleisten. Sobald der UiPath Orchestrator den Aufruf erhält, wird ein Warteschlangenobjekt angelegt. Die Automation Control Center Schicht verarbeitet Prozesse in einer Warteschlange, um die begrenzte Anzahl an Robotern optimal nutzen zu können. Dadurch wird verhindert, dass Prozesse gleichzeitig gestartet werden und zu Systemabstürzen führen. Die Priorisierung von Prozessen ist ein integraler Bestandteil, wobei zehn Prioritätsstufen zur Verfügung stehen, von „kritisch“ bis „niedrig“. Dies ermöglicht eine effiziente und gesteuerte Bearbeitung der Aufgaben. Sobald ein Prozess ausgeführt wird, erfolgt die Rückgabe der Ergebnisse an den Orchestrator, der die Informationen an Flowable weiterleitet. Durch diese Rückmeldung kann der Workflow fortgesetzt werden. Weiterhin sind Parallelisierungen innerhalb des Modells möglich, um die Effizienz der Prozesse zu maximieren. Abbildung 4 zeigt den Aufbau der Microservice-Architektur für wertgenerierende Microservices.

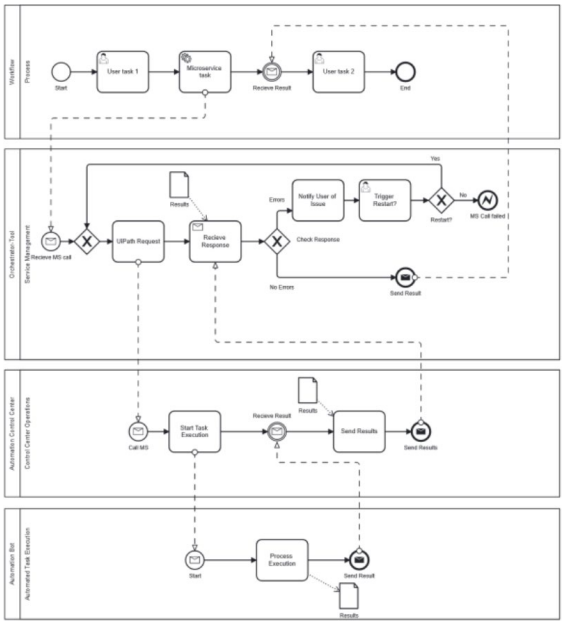


Abbildung 4 Wertgenerierende Microservice-Architektur Quelle: Eigene Darstellung

Der zweite Teil der Microservice-Architektur (Abbildung 5) konzentriert sich auf die Bereitstellung nicht-wertgenerierender Microservices in Form von wiederverwendbaren Modulen. Diese Module erfüllen unterstützende Funktionen, die zwar keinen direkten Mehrwert schaffen, aber für die Umsetzung von Geschäftsprozessen notwendig sind wie z.B. System-Authentifizierungen. Eine effiziente Implementierung dieser Module lässt sich durch den Einsatz von UiPath-Bibliotheken realisieren, die eine standardisierte Bereitstellung solcher Prozesse ermöglichen und somit zur Konsistenz und Wiederverwendbarkeit in der gesamten Architektur beitragen. Die Architektur der Bibliotheken besteht aus zwei zentralen Bereichen: der Entwicklung und der Aktualisierung. Zunächst werden die Module als eigenständige Prozesse innerhalb einer Bibliothek entwickelt, wobei jedes Modul eine spezifische Funktionalität abbildet, die es in den Geschäftsprozessen erfüllen soll. Ein Beispiel hierfür ist ein Authentifizierungsprozess, der als eigenständiges Modul in eine Bibliothek integriert wird. Diese Modularisierung ermöglicht eine flexible Wiederverwendung der Prozesse in unterschiedlichen Kontexten. Nach der Entwicklung werden die Module umfassend getestet, um sicherzustellen, dass sie fehlerfrei funktionieren. Anschließend wird die Bibliothek veröffentlicht, sodass die Module in weiteren Prozessen importiert und verwendet werden können. Um eine flexible Handhabung der Module zu gewährleisten, lassen sich Parameter wie Argumente, Assets oder Dropdown-Menüs definieren, durch die notwendige Informationen dynamisch in das Modul übergeben werden. Dies erhöht die Flexibilität bei der Anwendung der Module, da sie ohne Änderungen an ihrer Struktur in

verschiedenen Prozessen genutzt werden können. Der zweite wesentliche Aspekt der Architektur betrifft den Aktualisierungsprozess der Bibliotheken. Im RPA-Umfeld können selbst kleine Änderungen an Systemoberflächen zu Fehlern führen, weshalb eine regelmäßige Anpassung und Wartung der Module erforderlich ist. Um dies effizient zu gestalten, müssen sich Bibliotheken einfach aktualisieren lassen. Dazu wird das Modul in der Bibliothek angepasst, erneut getestet und nach erfolgreicher Prüfung erneut veröffentlicht. Anschließend erfolgt die Aktualisierung aller Prozesse, welche die Bibliothek nutzen, mithilfe eines Massenupdatetools. Dieses Tool ermöglicht es, alle betroffenen Prozesse automatisch auf die neueste Version der Bibliothek zu aktualisieren, wodurch der manuelle Aufwand und die Fehleranfälligkeit erheblich reduziert werden. Insgesamt ermöglicht die Bereitstellung und Verwaltung wiederverwendbarer Module in einer Microservice-Architektur eine hohe Flexibilität und Effizienz. Durch die klare Trennung zwischen wertgenerierenden und unterstützenden Prozessen wird die Architektur modularer und leichter wartbar. Die Verwendung von Bibliotheken spielt hierbei eine zentrale Rolle, da sie nicht nur die Wiederverwendbarkeit und Standardisierung fördern, sondern auch die unkomplizierte Wartung und Aktualisierung von Prozessen in dynamischen Umgebungen sicherstellen.

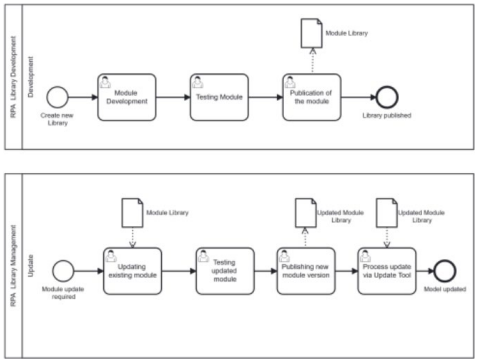


Abbildung 5 Nicht-wertgenerierende Microservice Architektur Quelle: Eigene Darstellung

Bewertung der Use Case Architektur

Für die Evaluierung der Microservice-Architektur wird das visuelle Hilfsmittel der Harvey Balls verwendet. Diese ermöglichen eine differenzierte Darstellung des Erfüllungsgrads der zuvor erhobenen Anforderungen. Ein leerer Kreis (○) steht für Nichterfüllung, ein vollständig ausgefüllter Kreis (●) steht für die vollständige Erfüllung. Zwischen diesen beiden Extremen ermöglichen die Abstufungen von 25 % (◔), 50 % (◑) und 75 % (◕) eine feinere Differenzierung der Erfüllungsgrade an die Architektur.

Tabelle 2 Harvey Balls Skala Quelle: Eigene Darstellung

Skala	Erfüllungsgrad	Beschreibung
○	*Nicht erfüllt (0 %)*	Die Architektur bietet aktuell keine Unterstützung, um diese spezifische Anforderung zu erfüllen.
◔	*Ansatzweise erfüllt (25 %)*	Es existieren initiale Eigenschaften oder Komponenten in der Architektur, durch welche die Anforderung in einem grundlegenden Umfang erfüllt werden, jedoch sind weitere Entwicklungen und Optimierungen notwendig.
◑	*Teilweise erfüllt (50 %)*	Die Architektur erfüllt diese Anforderung in einigen, jedoch nicht allen Aspekten. Die aktuelle Lösung adressiert die grundlegenden Bedürfnisse, doch es besteht noch erheblicher Bedarf an Verbesserungen.
◕	*Überwiegend erfüllt (75 %)*	Die Architektur erfüllt die Anforderung in den meisten Punkten effektiv. Kleinere Optimierungen oder Erweiterungen könnten jedoch noch implementiert werden, um eine vollständige und umfassendere Lösung zu bieten.
●	*Vollständig erfüllt (100 %)*	Die Architektur erfüllt die Anforderung vollständig und zuverlässig mit allen notwendigen Funktionen, die zur optimalen Erfüllung benötigt werden. Keine weiteren Verbesserungen sind erforderlich.

Nachfolgend werden die einzelnen Bewertungen kurz erläutert:

Wiederverwendbarkeit (●): Die Architektur unterstützt konkret zwei Arten von Wiederverwendbarkeit. Zum einen die Wiederverwendbarkeit in Form von Bibliotheken, zum anderen die Wiederverwendbarkeit von ganzen Services. Hierdurch unterstützt die vorgestellte Architektur die Anforderung der Wiederverwendbarkeit vollumfänglich.

Wartbarkeit (◕): Die vorliegende Architektur erleichtert die Wartbarkeit der Prozesse. Zukünftig muss nur noch an einer Stelle die Bibliothek oder der Service geändert werden, anstatt jeden Prozess einzeln anzupassen. Im Kontext von Bibliotheken ist zwar weiterhin die Verwendung eines Update Tools notwendig, dennoch wird die Wartbarkeit durch die Architektur deutlich verbessert.

Skalierbarkeit (◕): Die Skalierbarkeit wird durch die Verwendung von Warteschlangen optimiert, dennoch ist die Skalierbarkeit primär im UiPath Kontext durch die begrenzte Roboterverfügbarkeit bzw. durch das Lizenzmodell weitestgehend eingeschränkt.

Ersetzbarkeit (●): Die Architektur besitzt durch das Konzept der Microservices eine adäquate Ersetzbarkeit. Es ist möglich einzelne Services in der Flowable Prozesskette oder Bibliotheksmodule innerhalb der Ui-Path Prozesskette auszutauschen. Die einzige Voraussetzung für einen anpassungsfreien Austausch ist, dass Input und Output identisch sind.

Flexibilität (◕): Die Architektur zeigt sich anpassungsfähig durch die Unterscheidung zwischen wertgenerierenden und nicht-wertgenerierenden Microservices, sowie durch die Verwendung eines Orchestrierungstools wie Flowable. Diese Strukturierung ermöglicht es, zukünftig auch Microservices anderer Systeme zu integrieren, was die Flexibilität weiter steigert. Es ist allerdings zu beachten, dass starre/feste Prozessabläufe und zu spezifisch gestaltete Prozesse die Flexibilität einschränken können.

Testbarkeit (◑): Die Architektur fördert und erweitert die Testbarkeit, indem sie das separate Testen einzelner Module ermöglicht. Dies bedeutet, dass man jeden Microservice individuell und isoliert auf seine Funktionalität prüfen kann, was zu einer hohen Testbarkeit führt. Jedoch ist unklar, wie Integrationstests also Test, die auf das Zusammenspiel zwischen den einzelnen Microservices

innerhalb des Systems abzielen, realisiert werden können. Dies erweist sich als relevant, da unterschiedliche Systeme innerhalb der Prozesse verwendet werden.

Performance (◑): Die Performance lässt sich nicht genau bewerten, da dies über einen längeren Zeitraum beobachtet und anhand der dadurch gewonnenen Daten bewertet werden sollte. Dennoch stellt die Verwendung von Warteschlangen innerhalb der Architektur eine praktikable Lösung für die Lastenverteilung dar, was auf eine gute Performance hindeuten kann.

Stabilität (◑): Die Stabilität der Architektur ist vorrangig an die Stabilität der Systeme und Prozesse gebunden, weshalb sich eine Bewertung ebenfalls als schwierig erweist. Die Architektur ist offen für Mechanismen, welche die Stabilität fördern, wie der bereits eingebaute Restart Mechanismus im Fehlerfall. Solche Mechanismen sind wichtig, um bei Ausfällen oder Fehlern die Funktionalität der Prozesse aufrecht zu erhalten und schnelle Wiederanläufe zu ermöglichen. Allerdings müssen weitere Mechanismen eingeführt werden, um die Stabilität sicherzustellen.

Fehlerbehebungsmöglichkeiten (◑): Die Fehlerbehebungsmöglichkeiten sind durch den implementierten Restart-Mechanismus innerhalb der Architektur berücksichtigt worden, was die Resilienz gegenüber Störungen erhöht. Diese Möglichkeiten sind jedoch in hohem Maße von den spezifischen Prozessen abhängig. Daraus folgt, dass ohne detaillierte Kenntnisse über die einzelnen Prozessabläufe eine vollständige Bewertung der Fehlerbehebungsfähigkeit nicht möglich ist.

Integrationsfähigkeit (●): Die Architektur bietet eine hohe Integrationsfähigkeit, was durch die klar definierte Schnittstelle zwischen dem Orchestrierungstool und der RPA-Umgebung gewährleistet wird. Zudem ermöglicht die modulare Gestaltung eine flexible Einbindung neuer Dienste und die Erweiterung bestehender Funktionalitäten ohne größere Änderungen an der Gesamtarchitektur. Die Offenheit der Architektur erleichtert die Integration von Drittanbieter-Software und die Anpassung an geänderte Geschäftsanforderungen. Allerdings setzt die vollständige Ausschöpfung der Integrationsfähigkeit eine tiefergehende Prüfung der Kompatibilität und eine fehlerfreie Konfiguration der einzelnen Komponenten und Schnittstellen voraus.

Time-to-Market (◑): Die Architektur unterstützt eine modulare Bauweise von Prozessen, was zu einer potenziellen Beschleunigung der Entwicklungszyklen beiträgt. Dies kann sich positiv auf die Time-to-Market auswirken und somit zu einem schnelleren Rollout neuer Funktionalitäten und geänderten Anforderungen führen. Allerdings hängt die Time-to-Market auch von Faktoren wie der verwendeten Schnittstelle zwischen UiPath und Flowable sowie dem Reifegrad der Entwicklungs- und Deployment-Prozesse ab, da diese ebenfalls eine signifikante Rolle spielen.

Compliance und Governance (◔): In ihrer grundlegenden Form bietet die Architektur keine expliziten Funktionen zur Unterstützung von Compliance und Governance wie Überwachung oder Protokollierung. Sie ist jedoch in der Gestaltung offen für die Integration solcher Mechanismen. Die im Kontext zu Erstellung der Architektur verwendeten Softwarelösungen UiPath und Flowable enthalten bereits eingebaute Governance- und Compliance-Funktionen, welche die Einhaltung betrieblicher Richtlinien und Standards unterstützen.

Administrierbarkeit (◕): Die Architektur zeichnet sich durch ihre klare Strukturierung und Modularität aus, was die Administration erleichtert. Durch die Verwendung von etablierten Tools wie UiPath und Flowable, die bereits über umfassende Verwaltungsoberflächen verfügen, wird die Administrierbarkeit gestärkt. Dennoch hängt eine effektive Administrierbarkeit von der Einrichtung entsprechen der Management- und Monitoring-Tools ab, die in der Lage sind, das System im operativen Betrieb zu unterstützen.

Sicherheit (◔): Die Sicherheit ist ein kritischer Aspekt der Architektur, welcher besondere Aufmerksamkeit erfordert. Durch die Verwendung modularer Komponenten können Sicherheitsmaßnahmen gezielt und spezifisch für jeden Microservice implementiert werden. Dies fördert eine Sicherheitsarchitektur, die sich an den individuellen Sicherheitsanforderungen jeder Komponente orientiert. Die aktuelle Architektur besitzt durch die verwendeten Systeme bereits Sicherheitsmechanismen für Authentifizierung, Autorisierung und Verschlüsselung. Allerdings besteht erhebliches Optimierungspotenzial, insbesondere wenn man im RPA-Bereich Systeme verwendet, die umfassende Rollen- und Berechtigungskonzepte erfordern. Es sollten zusätzliche Maßnahmen ergriffen werden, um die Sicherheit weiter zu verbessern und sicherzustellen, dass keine übermäßigen Rechte existieren, speziell bei der Automatisierung kritischer Systeme.

FAZIT

Der Artikel hat sich mit der Entwicklung einer Microservice-Architektur im RPA-Umfeld beschäftigt, um Vorteile wie bessere Wiederverwendbarkeit, erhöhte Skalierbarkeit, größere Ersetzbarkeit und verbesserte Flexibilität zu realisieren. Diese Aspekte tragen dazu bei, Entwicklungs- und Wartungszeiten zu reduzieren und somit die Agilität und Kosteneffizienz zu steigern.

Ein zentrales Ergebnis ist der Aufbau einer modularen Architektur, die durch die Integration von RPA-Microservices und ein Orchestrierungstool unterstützt wird. Die Katalogisierung der Microservices sowie die Unterscheidung zwischen wertgenerierenden und nicht-wertgenerierenden Services fördern die Übersicht und Wiederverwendbarkeit.

Handlungsempfehlung

Um die langfristige Effizienz und Skalierbarkeit der Architektur sicherzustellen, sollte diese regelmäßig evaluiert und an sich ändernde Anforderungen angepasst werden. Logging- und Monitoring-Konzepte sind ebenfalls weiterzuentwickeln, insbesondere wenn die Anzahl an Microservices steigt. Ein weiterer Aspekt ist die kontinuierliche Überprüfung der eingesetzten Microservices, um unnötige Prozesse zu identifizieren und zusammenzuführen. Workshops und Schulungen sollten regelmä-

ßig angeboten werden, um das Wissen über Microservices im Unternehmen zu vertiefen. Die Optimierung der Schnittstellen zwischen den einzelnen Schichten, möglicherweise durch den Einsatz eines API-Gateways, ist ein zentraler Aspekt, um die Effizienz weiter zu steigern.

Ausblick

Die vorgestellte Architektur bildet die Grundlage für zukünftige Weiterentwicklungen. Durch die Einführung moderner Technologien wie künstlicher Intelligenz könnten Ausfälle proaktiv verhindert und Sicherheitsmaßnahmen verbessert werden. Langfristig könnte die Monetarisierung der entwickelten RPA-Microservices als Umsatzquelle dienen und Unternehmen als Innovationsführer in der Branche positionieren.

Literaturverzeichnis

Aguirre, Santiago; Rodriguez, Alejandro (2017): Automation of a Business Process Using Robotic Process Automation (RPA): A Case Study, S. 65–71. DOI: 10.1007/978-3-319-66963-2_7.

Alberth, Markus; Mattern, Michael (2017): Automation. Understanding robotic process Automation (RPA). In: *JOURNAL - THE CAPCO INSTITUTE JOURNAL OF FINANCIAL TRANSFORMATION* (46). Online verfügbar unter https://www.capco.com/-/media/CapcoMedia/Capco-2023/Capco-Institute/Journal-46/JOURNAL46_5_Alberth.ashx.

Alpers, Sascha; Becker, Christoph; Oberweis, Andreas; Schuster, Thomas (2015): Microservice Based Tool Support for Business Process Modelling. In: *2015 IEEE 19th International Enterprise Distributed Object Computing Workshop,* S. 71–78. DOI: 10.1109/EDOCW.2015.32

Al-Slais, Yaqoob; Ali, Mazen (2023): Robotic Process Automation and Intelligent Automation Security Challenges: A Review. In: *2023 International Conference On Cyber Management And Engineering (CyMaEn)*, S. 71–77. DOI: 10.1109/CyMaEn57228.2023.10050996.

Asatiani, Aleksandre; Penttinen Esko (2016): TURNING ROBOTIC PROCESS AUTOMATION INTO COMMERCIAL SUCCESS – CASE OPUSCAPITA. In: *Journal of Information Technology Teaching Cases* (6(2)), S. 67–74.

AWS (o. D.): Was ist eine API? – Anwendungsprogrammierschnittstelle? Hg. v. Amazon Web Services, Inc. Online verfügbar unter https://aws.amazon.com/de/what-is/api/, zuletzt geprüft am 11.01.2024.

Axmann, Bernhard; Harmoko, Harmoko (2020): Robotic Process Automation: An Overview and Comparison to Other Technology in Industry 4.0. In: *10th International Conference on Advanced Computer Information Technologies (ACIT)*, S. 559–562. DOI: 10.1109/ACIT49673.2020.9208907.

Baškarada, Saša; Nguyen, Vivian; Koronios, Andy (2020): Architecting Microservices: Practical Opportunities and Challenges. In: *Journal of Computer Information Systems* 60 (5), S. 428–436. DOI: 10.1080/08874417.2018.1520056.

Berruti, Federico; Nixon, Graeme; Taglioni, Giambattista; Whiteman, Rob (2017): Intelligent process automation: The engine at the core of the next-generation operating model. In: *McKinsey & Company*, 2017. Online verfügbar unter https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/intelligent-process-automation-the-engine-at-the-core-of-the-next-generation-operating-model, zuletzt geprüft am 11.01.2024.

Blinowski, Grzegorz; Ojdowska, Anna; Przybylek, Adam (2022): Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation. In: *IEEE Access* 10, S. 20357–20374. DOI: 10.1109/ACCESS.2022.3152803.

Brettschneider, Jennifer (2020): Bewertung der Einsatzpotenziale und Risiken von Robotic Process Automation. In: *HMD* 57 (6), S. 1097–1110. DOI: 10.1365/s40702-020-00621-y.

Bygstad, Bendik (2017): Generative Innovation: A Comparison of Lightweight and Heavyweight IT. In: *Journal of Information Technology* 32 (2), S. 180–193. DOI: 10.1057/jit.2016.15.

Celonis (o. D.): Unser Unternehmen. Hg. v. Celonis. Online verfügbar unter https://www.celonis.com/de/company/, zuletzt geprüft am 10.03.2024.

Choi, Daehyoun; R'bigui, Hind; Cho, Chiwoon (2021): Candidate Digital Tasks Selection Methodology for Automation with Robotic Process Automation. In: *Sustainability* 13 (16), S. 8980. DOI: 10.3390/su13168980.

Choi, Daehyoun; R'bigui, Hind; Cho, Chiwoon (2022): Enabling the Gab Between RPA and Process Mining: User Interface Interactions Recorder. In: *IEEE Access* 10, S. 39604–39612. DOI: 10.1109/ACCESS.2022.3165797.

Da Costa, Diogo António Silva; Mamede, Henrique São; Da Mira Silva, Miguel (2022): Robotic Process Automation (RPA) Adoption: A Systematic Literature Review. In: *Engineering Management in Production and Services* 14 (2), S. 1–12. DOI: 10.2478/emj-2022-0012.

Dey, Sourav; Das, Arindam (2019): Robotic process automation: assessment of the technology for transformation of business processes. In: *IJBPIM* 9 (3), Artikel 100927, S. 220–230. DOI: 10.1504/IJBPIM.2019.100927.

Doguc, Ozge (2020): Robot Process Automation (RPA) and Its Future. In: Ümit Hacıoğlu (Hg.): Handbook of research on strategic fit and design in business ecosystems. Hershey, PA, USA: IGI Global Business Science Reference (Advances in E-Business Research (AEBR) book series), S. 469–492. Online verfügbar unter https://www.researchgate.net/profile/Ozge-Doguc-2/publication/338302068_Robot_Process_Automation_RPA_and_Its_Future/links/5f5772f592851c250b9d23ad/Robot-Process-Automation-RPA-and-Its-Future.pdf, zuletzt geprüft am 10.12.2023.

Dowalil, Herbert (2018): Grundlagen des modularen Softwareentwurfs. Der Bau langlebiger Mikro- und Makro-Architekturen wie Microservices und SOA 2.0. München: Hanser.

Drawehn, Jens; Krause, Tobias; Renner, Thomas; Kintz, Maximilien (2022): Robotic Process Automation in Versicherungsunternehmen. Erfahrungen und Best Practices beim Einsatz von RPA: Fraunhofer-Gesellschaft. Online verfügbar unter https://www.digital.iao.fraunhofer.de/content/dam/iao/ikt/de/documents/RPA_in_Versicherungsunternehmen.pdf, zuletzt geprüft am 13.12.2023.

Eyerman, Stijn; Hur, Ibrahim (2022): Efficient Asynchronous RPC Calls for Microservices: DeathStarBench Study. DOI: 10.48550/arXiv.2209.13265.

Farley, David (o. D.): Modernes Software Engineering - Bessere Software schneller und effektiver entwickeln by David Farley. Hg. v. O'Reilly. Online verfügbar unter https://www.oreilly.com/library/view/modernes-software-engineering/9783747506363/Text/k10.html, zuletzt geprüft am 20.01.2024.

Fowler, Martin (2020): Domain Driven Design. Hg. v. martinFowler.com. Online verfügbar unter https://martinfowler.com/bliki/DomainDrivenDesign.html, zuletzt geprüft am 15.04.2024.

Fowler, Martin; Lewis, James (2014): Microservices. a definition of this new architectural term. Hg. v. martinFowler.com. Online verfügbar unter https://martinfowler.com/articles/microservices.html?source=post_page, zuletzt aktualisiert am 25.03.2014, zuletzt geprüft am 16.11.2023.

Gartner (o. D.a): Definition of Architecture - Gartner Information Technology Glossary. Hg. v. Gartner. Online verfügbar unter https://www.gartner.com/en/information-technology/glossary/architecture#:~:text=IT%20architecture%20is%20a%20series%20of%20principles%2C%20guidelines,communications%2C%20development%20methodologies%2C%20modeling%20tools%20and%20organizational%20structures., zuletzt geprüft am 19.01.2024.

Gartner (o. D.b): Definition of Robotic Process Automation. Hg. v. Gartner. Online verfügbar unter https://www.gartner.com/en/information-technology/glossary/robotic-process-automation-software, zuletzt aktualisiert am 11.12.2023, zuletzt geprüft am 11.12.2023.

GitLab (2022): What are the benefits of a microservices architecture? Hg. v. GitLab. Online verfügbar unter https://about.gitlab.com/blog/2022/09/29/what-are-he-benefits-of-a-microservices-architecture/, zu- letzt aktualisiert am 29.09.2022, zuletzt geprüft am 23.01.2024.

Habibullah, Safa; Liu, Xiaodong; Tan, Zhiyuan; Zhang, Yonghong; Liu, Qi (2019): Reviving Legacy Enterprise Systems with Micro service-Based Architecture with in Cloud Environments. In: *8th International Conference on Soft Computing, Artificial Intelligence and Applications* 9, S. 173–186. DOI: 10.5121/csit.2019.90713.

Hanussek, Marc (2019): RPA meets KI oder: wie intelligente Softwareroboter Ihre Prozesse automatisieren. Hg. v. Fraunhofer IAO - BLOG. Online verfügbar unter https://blog.iao.fraunhofer.de/rpa-meets-ki-oder-wie-intelligente-softwareroboter-ihre-prozesse-automatisieren/, zuletzt aktualisiert am 06.07.2021, zuletzt geprüft am 28.12.2023.

Hu, Chenglie (2023): An Introduction to Software Design. Concepts, Principles, Methodologies, and Techniques. 1st ed. 2023. Cham: Springer International Publishing; Imprint Springer. Online verfügbar unter https://link.springer.com/book/10.1007/978-3-031-28311-6.

IEEE Computer Society/Software & Systems Engineering Standards (2022): IEEE/ISO/IEC International Standard for Software, systems and enterprise--Architecture description. DOI: 10.1109/IEEESTD.2022.9938446.

Indrasiri, Kasun; Siriwardena, Prabath (2018): Microservices for the Enterprise. Designing, Developing, and Deploying. 1st ed. 2018. New York: Apress (Springer eBook Collection). Online verfügbar unter https://link.springer.com/content/pdf/10.1007/978-1-4842-3858-5.pdf.

Institute for Robotic Process Automation & Artificial Intelligence (o. D.): What is Robotic Process Automation? | IRPAAI. Hg. v. IRPA AI. Online verfügbar unter https://irpaai.com/what-is-robotic-process-automation/, zuletzt aktualisiert am 11.12.2023, zuletzt geprüft am 11.12.2023.

Ivančić, Lucija; Suša Vugec, Dalia; Bosilj Vukšić, Vesna (2019): Robotic Process Automation: Systematic Literature Review 361, S. 280–295. DOI: 10.1007/978-3-030-30429-4_19.

Jamshidi, Pooyan; Pahl, Claus; Mendonca, Nabor C.; Lewis, James; Tilkov, Stefan (2018): Microservices: The Journey So Far and Challenges Ahead. In: IEEE Softw. 35 (3), S. 24–35. DOI: 10.1109/MS.2018.2141039.

Kalske, Miika; Mäkitalo, Niko; Mikkonen, Tommi (2018): Challenges When Moving from Monolith to Microservice Architecture. In: Irene Garrigós und Manuel Wimmer (Hg.): Current Trends in Web Engineering. ICWE 2017 International Workshops, Liquid Multi-Device Software and EnWoT, practi-O-web, NLPIT, SoWeMine ; Rome, Italy, June 5-8, 2017 ; revised selected papers, Bd. 10544. Cham: Springer International Publishing (Lecture Notes in Computer Science, 10544), S. 32–47. Online verfügbar unter https://link.springer.com/chapter/10.1007/978-3-319-74433-9_3.

Karnowski, Veronika (2013): Diffusionstheorie. In: Wolfgang Schweiger und Andreas Fahr (Hg.): Handbuch Medienwirkungsforschung. Wiesbaden: Springer VS, S. 513–528. Online verfügbar unter https://link.springer.com/chapter/10.1007/978-3-531-18967-3_27.

Khan, Ovais; Siddiqui, Nabil; Oleson, Timothy; Fussell, Mark (2021): Embracing Microservices Design. A practical guide to revealing anti-patterns and architectural pitfalls to avoid microservices fallacies. 1st edition. Erscheinungsort nicht ermittelbar, Boston, MA: Packt Publishing; Safari.

Kirchmer, Mathias (2017): Robotic Process Automation - Pragmatic Solution or Dangerous Illusion? In: *BTOES Insights (Business Transformation and Operational Excellence Summit Insights)*. Online verfügbar unter https://www.researchgate.net/publication/317730848_Robotic_Process_Automation_-_Pragmatic_Solution_or_Dangerous_Illusion, zuletzt geprüft am 04.01.2024.

Koch, Oliver; Wildner, Stephan (2020): Intelligent Robotic Process Automation. Konzeption eines Ordnungsrahmens zur Nutzung künstlicher Intelligenz für die Prozessautomatisierung. In: Rüdiger Buchkremer, Thomas Heupel und Oliver Koch (Hg.): Künstliche Intelligenz in Wirtschaft & Gesellschaft. Auswirkungen, Herausforderungen & Handlungsempfehlungen. Wiesbaden, Heidelberg: Springer Gabler (FOM-Edition), S. 211–230. Online verfügbar unter https://link.springer.com/book/10.1007/978-3-658-29550-9, zuletzt geprüft am 04.01.2024.

Köhler-Schute, Christiana (2020): Robotic Process Automation in Unternehmen. Praxisorientierte Methoden und Vorgehensweisen zur Umsetzung von RPA-Initiativen. Berlin: KS-Energy-Verlag.

Kokina, Julia; Blanchette, Shay (2019): Early evidence of digital labor in accounting: Innovation with Robotic Process Automation. In: *International Journal of Accounting Information Systems* 35, S. 100431. DOI: 10.1016/j.accinf.2019.100431.

Kroll, Christian; Bujak, Adam; Darius, Volker; Enders, Wolfgang; Esser, Marcus (2016): Robotic Process Automation - Robots conquer business processes in back offices. A 2016 study conducted by Capgemini Consulting and Capgemini Business Services. Hg. v. Capgemini. Online verfügbar unter https://www.capgemini.com/consulting-de/wp-content/uploads/sites/32/2017/08/robotic-process-automation-study.pdf, zuletzt geprüft am 04.01.2023.

Lal Sahni, Dhanik (2023): What is Microservice Architecture? Hg. v. Salesforcecodex. Online verfügbar unter https://stories.salesforcecodex.com/2023/05/salesforce/what-is-microservice-architecture/, zuletzt aktualisiert am 17.05.2023, zuletzt geprüft am 19.11.2023.

Langmann, Christian; Turi, Daniel (2021): Robotic Process Automation (RPA) - Digitalisierung und Automatisierung von Prozessen. Voraussetzungen, Funktionsweise und Implementierung am Beispiel des Controllings und Rechnungswesens. 2. Auflage. Wiesbaden, Heidelberg: Springer Gabler.

Lhuer, Xavier (2016): The next acronym you need to know about: RPA (robotic process automation. Hg. v. McKinsey. Online verfügbar unter https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-next-acronym-you-need-to-know-about-rpa, zuletzt geprüft am 13.05.2024.

Madakam, Somayya; Holmukhe, Rajesh M.; Kumar Jaiswal, Durgesh (2019): The Future Digital Work Force: Robotic Process Automation (RPA). In: *JISTEM* 16, S. 1–17. DOI: 10.4301/S1807-1775201916001.

Manning, Louise (2020): Moving from a compliance-based to an integrity-based organizational climate in the food supply chain. In: *Comprehensive reviews in food science and food safety* 19 (3). DOI: 10.1111/1541-4337.12548.

Microsoft Learn (o. D.): Microservice-Architekturstil. Hg. v. Microsoft Learn. Online verfügbar unter https://learn.microsoft.com/de-de/azure/architecture/guide/architecture-styles/microservices, zuletzt aktualisiert am 19.11.2023, zuletzt geprüft am 19.11.2023.

Microsoft Learn (2023): Entwerfen einer an Microservice orientierten Anwendung. Hg. v. Microsoft Learn. Online verfügbar unter https://learn.microsoft.com/de-de/dotnet/architecture/microservices/multi-container-microservice-net-applications/microservice-application-design, zuletzt aktualisiert am 10.05.2023, zuletzt geprüft am 19.11.2023.

Moffitt, Kevin C.; Rozario, Andrea M.; Vasarhelyi, Miklos A. (2018): Robotic Process Automation for Auditing. In: *Journal of Emerging Technologies in Accounting* 15 (1), S. 1–10. DOI: 10.2308/jeta-10589.

Montemagno, James; Warren, Genevieve; Jain, Tarun; Coulter, David; Veloso, Miguel et al. (2022): Communication in a microservice architecture. Hg. v. Microsoft Learn. Online verfügbar unter https://learn.microsoft.com/en-us/dotnet/architecture/microservices/architect-microservice-contai-nerapplications/communication-in-microservicearchitecture, zuletzt aktualisiert am 21.01.2024, zuletzt geprüft am 21.01.2024.

Nadareishvili, Irakli; Mitra, Ronnie; McLarty, Matt; Amundsen, Michael (2016): Microservice architecture. Aligning principles, practices, and culture. First Edition, Second Release. Beijing, Boston, Farnham, Sebastopol, Tokyo: O´Reilly.

Newman, Sam (2020): Vom Monolithen zu Microservices. Patterns, um bestehende Systeme Schritt für Schritt umzugestalten. Heidelberg: O'Reilly.

Newman, Sam (2021): Building microservices. Designing fine-grained systems. Second edition. Beijing, Boston, Farnham, Sebastopol, Tokyo: O'Reilly Media.

Niedermaier, Sina; Koetter, Falko; Freymann, Andreas; Wagner, Stefan (2019): On Observability and Monitoring of Distributed Systems – An Industry Interview Study. In: Sami Yangui, Ismael Bouassida Rodriguez, Khalil Drira, Zahir Tari und Pagination_Cover (Hg.): Service-Oriented Computing. 17th International Conference, ICSOC 2019, Toulouse, France, October 28–31, 2019, Proceedings, Bd. 11895. 1st ed. 2019. Cham: Springer (Springer eBooks Computer Science, 11895), S. 36–52. Online verfügbar unter https://link.springer.com/chap- ter/10.1007/978-3-030-33702-5_3.

Průcha, Petr; Skrbek, Jan (2022): API as Method for Improving Robotic Process Automation. In: Andrea Marrella, Raimundas Matulevičius, Renata Gabryelczyk, Bernhard Axmann, Vesna Bosilj Vukšić, Walid Gaaloul et al. (Hg.): Business Process Management: Blockchain, Robotic Process Automation, and Central and Eastern Europe Forum. BPM 2022 Blockchain, RPA, and CEE Forum, Münster, Germany, September 11–16, 2022, Proceedings. 1st ed. 2022. Cham: Springer International Publishing; Imprint Springer (Lecture Notes in Business Information Processing, 459). Online verfügbar unter https://link.springer.com/chapter/10.1007/978-3-031-16168-1_17.

PWC South Africa (o. D.): Robotic process automation. Hg. v. PWC South Africa. Online verfügbar unter https://www.pwc.co.za/en/services/consulting/robotic-process-automation.html, zuletzt geprüft am 11.12.2023.

RedHat (2019): Wie funktioniert ein API-Gateway? Hg. v. RedHat. Online verfügbar unter https://www.redhat.com/de/topics/api/what-does-an-api-gateway-do, zuletzt geprüft am 07.12.2023.

Santos, Filipa; Pereira, Rúben; Vasconcelos, José Braga (2020): Toward robotic process automation implementation: an end-to-end perspective. In: *BPMJ* 26 (2), S. 405–420. DOI: 10.1108/bpmj-12-2018-0380.

SAP (o. D.): Was ist Prozessautomatisierung? Hg. v. SAP. Online verfügbar unter https://www.sap.com/germany/products/technology-platform/process-automation/what-is-process-automation.html, zuletzt geprüft am 10.12.2023.

Shidaganti, Ganeshayya; Salil, Sreya; Anand, Prarthana; Jadhav, Vaishnavi (2021): Robotic Process Automation with AI and OCR to Improve Business Process: Review. In: *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, S. 1612–1618. DOI: 10.1109/ICESC51422.2021.9532902.

Singleton, Andy (2016): The Economics of Microservices. In: IEEE Cloud Comput. 3 (5), S. 16–20. DOI: 10.1109/MCC.2016.109.

Syed, Rehan; Suriadi, Suriadi; Adams, Michael; Bandara, Wasana; Leemans, Sander J.J.; Ouyang, Chun et al. (2020): Robotic Process Automation: Contemporary themes and challenges. In: *Computers in Industry* 115, S. 103162. DOI: 10.1016/j.compind.2019.103162.

Taibi, Davide; Lenarduzzi, Valentina; Pahl, Claus (2017): Processes, Motivations, and Issues for Migrating to Microservices Architectures: An Empirical Investigation. In: *IEEE Cloud Comput.* 4 (5), S. 22–32. DOI: 10.1109/MCC.2017.4250931.

UiPath (o. D.): KI und RPA – die nächste Stufe der Automatisierung | UiPath. Hg. v. UiPath. Online verfügbar unter https://www.uipath.com/de/automation/ai-and-rpa, zuletzt geprüft am 28.12.2023.

van der Aalst, Wil M. P.; Bichler, Martin; Heinzl, Armin (2018): Robotic Process Automation. In: *Bus Inf Syst Eng* 60 (4), S. 269–272. DOI: 10.1007/s12599-018-0542-4.

Vitharanage, Imesha; Thibbotuwawa, Amila (2021): Enterprise Robotic Process Automation. In: *BPRM* 01 (01), S. 10–12. DOI: 10.31705/BPRM.2021.2.

Wanner, Jonas; Hofmann, Adrian; Fischer, Marcus; Janiesch, Christian; Imgrund, Florian; Geyer-Klingebert, Jerome (2019): Process Selection in RPA Projects – Towards a Quantifiable Method of Decision Making. In: *Fortieth International Conference on Information Systems*. Online verfügbar unter https://opus.bibliothek.uni-augsburg.de/opus4/frontdoor/deliver/index/docId/95923/file/95923.pdf, zuletzt geprüft am 06.01.2024.

Willcocks, Leslie; Lacity, Mary; Craig, Andrew (2015): The IT Function and Robotic Process Automation. In: *London School of Economics and Political Science*. Online verfügbar unter https://eprints.lse.ac.uk/64519/1/OUWRPS_15_05_published.pdf, zuletzt geprüft am 10.12.2023.

Wolff, Eberhard (2018): Microservices. Grundlagen flexibler Softwarearchitekturen. 2., aktualisierte Auflage. Heidelberg: dpunkt.verlag.

Yousif, Mazin (2016): Microservices. In: *IEEE Cloud Comput.* 3 (5), S. 4–5. DOI: 10.1109/MCC.2016.101.

Zhang, Chanyuan; Thomas, Chanta; Vasarhelyi, Miklos A. (2021): Attended Process Automation in Audit: A Framework and A Demonstration. In: Journal of Information Systems 36 (2). DOI: 10.2308/ISYS-2020-073.

SAP GUI vs. SAP Fiori: A Brief Practical Comparison

Patrick E.-A. Möbert

Faculty of Computer Science
and Mathematics
Hochschule München,
University of Applied Sciences
Munich, Bavaria, Germany
moebert@hm.edu

Abstract

Many SAP users express frustration with the user interface and the complexity of data entry. Over the years, SAP has implemented various approaches to enhance user interaction and the overall user experience of its software products. This raises the question of whether these intentions truly address the users' concerns. Therefore, this paper compares the SAP GUI and SAP Fiori, highlighting the respective advantages and disadvantages of both frontends. The analysis shows that SAP Fiori offers significant benefits, especially for newly developed applications. In particular, SAP Fiori visualizes the relationships between business process steps – previously represented as separate transactions – by clearly depicting the underlying business process and its associated document flow, and it offers more effective error handling functionalities. These features can be especially valuable for first-time users and in university education, as it helps users better understand complex business processes. However, professional users with years of experience may prefer to continue using the SAP GUI, as they are accustomed to its function-oriented, transactional concept and familiar design.

Keywords

SAP GUI, SAP S/4HANA, SAP Fiori, User Experience, User Interaction, UX, UI

1. Introduction

SAP and its ERP systems have been an integral part of university curricula and training programs for decades, ever since the company was founded in 1972. However, conveying knowledge about business processes and how they are mapped and executed within SAP systems has always been a challenge [1]. To address this, SAP has continuously sought to improve the usability and user experience of its applications, aiming to make their often complex functionalities more accessible and understandable.

SAP's most recent initiative in this regard is the introduction of its modern user interface, SAP Fiori, which was launched in 2013 as a replacement for the traditional SAP GUI [2]. SAP Fiori fundamentally shifted the usage concept from a function-oriented to a role-based approach, utilizing app tiles and other user-friendly elements [3]. Designed specifically as the new user frontend for SAP's latest ERP software version, SAP S/4HANA (introduced in 2015) [4], SAP Fiori is intended to make SAP systems significantly more intuitive and better aligned with contemporary user expectations [5]. Moreover, SAP aims to reduce the complexity of traditional SAP transactions and enable faster, more efficient, and less error-prone processing of business processes [6].

There are many similar comparisons of SAP GUI and SAP Fiori available online, typically published by SAP consulting firms and often based on product specifications provided by SAP Marketing or Product Management [7]. These comparisons usually highlight the conceptual differences but rarely take into account the practical characteristics of each interface. As a result, they offer limited value when it comes to evaluating whether the real needs of first-time SAP users are truly met. Consequently, this paper will discuss and compare the pros and cons of both user interfaces from a practical perspective, aiming to support an informed decision between the two technologies.

Other studies suggest that usability perception depends on software customization during the implementation phase as well as the quality of end-user training, which means that both the structure of training sessions and the competence of trainers can impact user acceptance of a software solution [8]. Consequently, preconfigured and inflexible user interfaces, along with overly complex explanations during training, are less advantageous for inexperienced users and their comprehension. Therefore, this investigation will focus on functions such as help features and simple business process representations, which specifically support the initial learning process of first-time SAP users and have not yet been reported elsewhere. These aspects play an important role in university education, as they help establish usability-based user acceptance and strengthen users' understanding. Nevertheless, this paper does not aim to provide a comprehensive overview of all new SAP Fiori functions [9].

This paper is structured as follows: First, the different versions of the most commonly used SAP GUIs for SAP's ERP systems will be explained. This is followed by an in-depth analysis of the help functions provided by SAP GUI and SAP Fiori, which can be particularly helpful when using new and complex software products. Sub-

sequently, selected application functions will be considered, with a focus on user experience and interaction. Finally, the discussion concludes with a brief summary.

This study is based on empirical investigations conducted from March to July 2025 using the SAP S/4HANA 2022 release.

2. SAP GUI Versions

SAP has developed and provided up to 30 different GUIs for its various software products over time. The most popular is the classic SAP GUI for Windows, which has been in use with SAP R/3 since the early 1990s. An exclusive GUI version for Apple Macintosh computers was developed in the late 1990s, but it was never officially released. The official SAP GUI for Java, designed for non-Windows operating systems such as Mac OS X (today macOS), Linux, and others, was introduced in 2007 [10]. The first web-based SAP GUI for HTML, based on the Internet Transaction Server (ITS), was released in 2001 and enabled access to SAP R/3 via a web browser. The HTML GUI can be launched from the Windows GUI using the transaction code '/nWEBGUI'. However, the HTML GUI does not offer the full functionality and user experience of the Windows GUI, as certain features and integrations are limited or unavailable in the browser-based version.

As previously mentioned, SAP Fiori was introduced in 2013 and is built on the SAPUI5 UI framework, which utilizes the web standards HTML5, CSS, and JavaScript. In contrast to the classic client-server model, SAP Fiori includes, in addition to the backend server, a frontend server that provides the user interface for the browser. The business logic remains in the backend, and communication takes place via OData services. SAP Fiori can be initiated from the traditional GUIs via transaction code /n/UI2/FLP.

3. Help Functions

SAP has managed to develop several highly successful software products in the past. Beyond their extensive functional scope, there are several key factors that have contributed to this success. Some general aspects of using a software product can make for a pleasant user experience. On the other hand, there are also common pitfalls that users may find dissatisfying when they encounter them.

Almost as a matter of course, a simple hourglass icon indicates system activity and wait times, preventing users from clicking buttons or links multiple times. The hourglass was introduced by SAP with the graphical user interface as early as the early 1990s.

Another important yet often underestimated feature is the help function that supports users whenever they encounter uncertainty or need clarification. Nowadays, this is typically achieved with an integrated help tool or a separate webpage that provides users with further information on particular fields or topics, as well as answers to their questions. An example is SAP, which provides its online help through the SAP Help Portal, accessible at help.sap.com.

Fig. 1 illustrates how the incorporated SAP F1 help appears in the SAP GUI. It is referred to as F1 help because an additional window – labeled "Performance Assistant" in this SAP version – opens when the cursor is placed in the relevant field (the plant field in this example) and the F1 key is pressed on the keyboard. Additionally, the F1 help can also be accessed in case of an error or warning message, although it may not provide additional information for every possible error or warning.

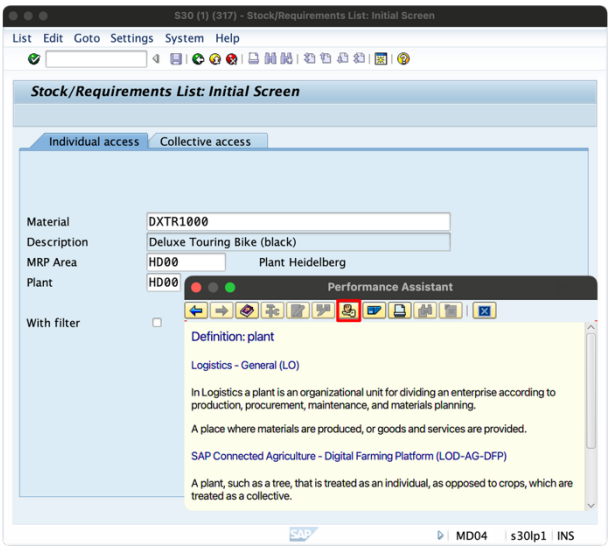


Fig. 1. Example of the help window displayed when accessing the SAP F1 help function, referred to as "Performance Assistant" [Screenshot from SAP S/4HANA, as of November 2025, © SAP SE. All rights reserved].

However, a feature within SAP that is probably not widely known is the ability to directly configure or change a missing or incorrect parameter. This can be done by navigating from the help window to the configuration site using the icon which displays a head/key symbol [see Fig. 1]. This function leads the user directly

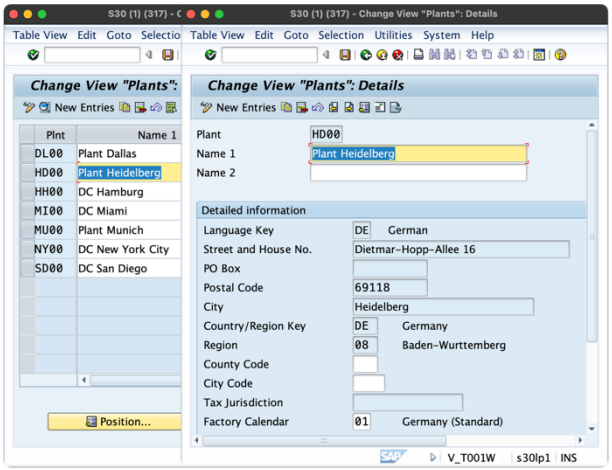


Fig. 2. Example of the windows that appear when navigating using the SAP F1 help function [Screenshot from SAP S/4HANA, as of November 2025, © SAP SE. All rights reserved].

to the location where a missing entry can be configured or a wrong specification can be corrected [Fig. 2].

This feature – which allows direct navigation to the configuration site of a parameter – is not always available and is probably seldom needed. However, it can be very useful for trainers or lecturers configuring specific areas of an SAP system in a training environment, or when something is missing, or an error occurs. Moreover, it demonstrates just how forward-thinking some of the SAP developers have been.

Fig. 3 illustrates the F1 help function similarly realized with SAP Fiori for this example.

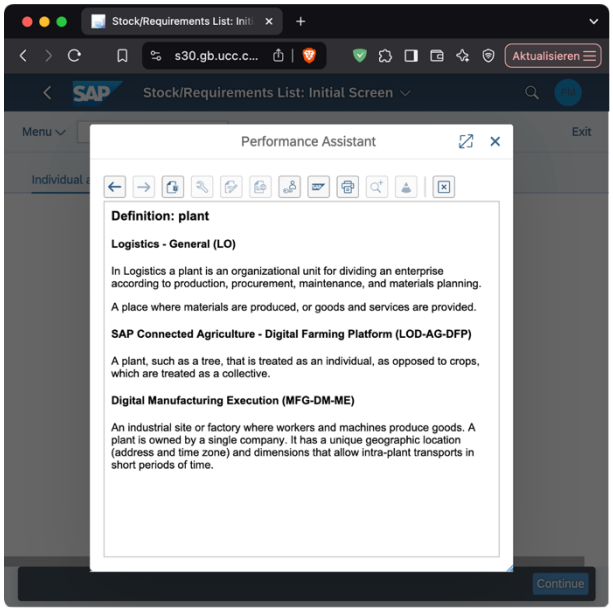
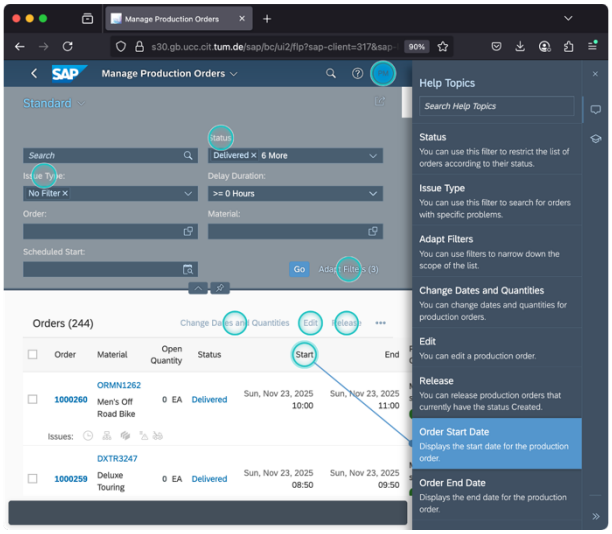


Fig. 3. Example of the F1 help window when accessed with SAP Fiori [Screenshot from SAP S/4HANA, as of November 2025, © SAP SE. All rights reserved].

The available functionality, including navigation to the configuration site, is identical to that in the SAP GUI, since both frontends essentially use the same underlying code for processing.



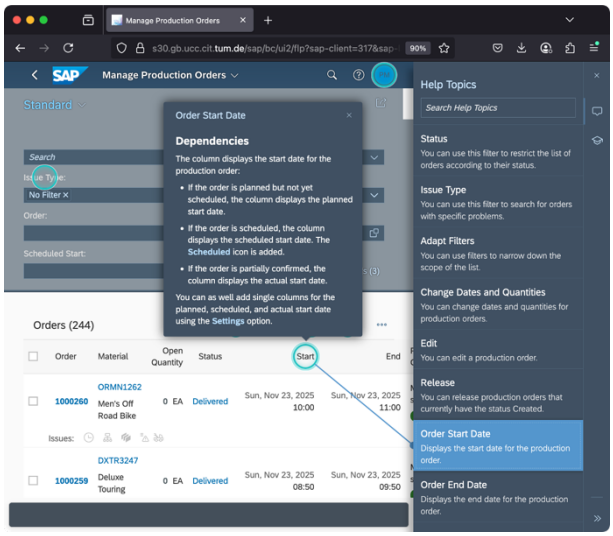


Fig. 5. Example of the new SAP Fiori F1 help function [Screenshot from SAP S/4HANA, as of November 2025, © SAP SE. All rights reserved].

A new and more advanced help feature is available in SAP Fiori, which can be especially beneficial for new SAP users. As shown in Fig. 4, when entering an app and pressing F1, an extended help sidebar appears on the right. This sidebar presents various help topics linked to specific fields, which are highlighted with blue circles. By hovering the mouse over these blue circles, users can access additional pop-up windows containing detailed information relevant to each field [Fig. 5]. This contextual help makes it easier for users to understand the application's features and functions more efficiently.

4. Application Functions

As stated above, the usability concept has been changed in SAP Fiori from function-oriented to role-based, which basically means that there are groupings of specific application functions (in the form of several tiles) that belong to a certain user role such as "Sales Representative". The SAP GUI user interaction was based on a tree structure (comparable to the sinistral tree of the Microsoft Windows Explorer), where functions have been grouped by application areas such as "Sales and Distribution" or "Materials Management" [see Fig. 6]. A specific function was selected by drill-down within the tree structure. In addition, it was possible to access a specific application function directly by entering a respective so-called transaction code in the OK-code field located in the upper left corner [Fig. 6]. A transaction code has been assigned to any application function, for instance '/nVA01' for transaction "Create Sales Order".

Considering the clarity of the different representations, the new concept of SAP Fiori might be advantageous for users having a certain professional role in a company related to the execution of up to about ten different business functions during daily work. Moreover, the application tiles used can be grouped and arranged user individually on the SAP Fiori desktop, whereas the

SAP GUI allows for the definition of favorite transactions located in the beginning of the tree structure instead [confer Fig. 6].

Regarding the design and look of the functions provided by the specific application tiles and transactions, the flat design of SAP Fiori apps seems to be not superior to the "old" transaction design of the SAP GUI, in particular when using the "Signature Design" screen theme provided in the GUI preferences (even compared to the "SAP Belize Deep" or any other appearance that can be chosen in the SAP Fiori settings). This will be rather obvious looking at the "Manage Business Partner" application as an example: Even though the SAP GUI transaction to maintain business partners is already quite complex, the SAP Fiori application seems to be even more perplexing. The application consists of a long list with dozens of information and data to be entered, and jump marks in form of tabstrips for specific topics can be used to directly navigate to the desired section of the list (such as address data or bank account information), making it difficult to visually recognize any of the information provided (especially for training classes). Hence, SAP Fiori offers no real advancement at this point and for all business applications.

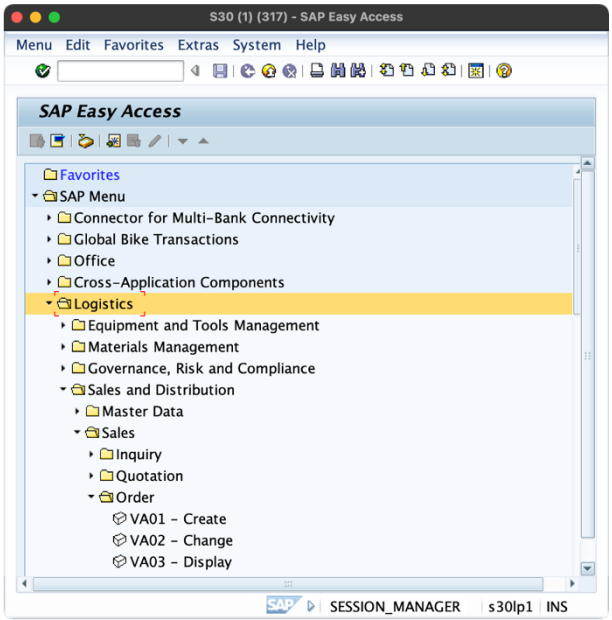


Fig. 6. Example of the SAP GUI frontend design [Screenshot from SAP S/4HANA, as of November 2025, © SAP SE. All rights reserved].

Sometimes, the new positioning and logic of buttons such as "Go", "Edit", "Create", "Save", or "Back" in SAP Fiori can be a bit confusing at first. However, button placement and behavior were not always consistent in the classic SAP GUI either, as these aspects were often determined by the individual developer. Only the logic for the "Back", "Exit", and "Cancel" buttons was implemented consistently across the SAP GUI.

Furthermore, not all applications or transactions have been newly developed for SAP Fiori or SAP S/4HANA. While several new SAP Fiori applications have been created, particularly for analytics and fact sheets, much of the underlying ABAP code has simply been transferred from SAP R/3 (specifically SAP ERP Central Component, SAP ECC) to SAP S/4HANA. For example, the "Create Sales Order" and "Monitor Stock / Requirements List" apps are essentially identical to their former SAP ECC counterparts. As a result, many of these transactions are displayed in SAP Fiori as HTML views, rather than as fully redesigned Fiori apps.

However, the main advantage of SAP Fiori is that it no longer requires a local SAP GUI installation. Instead, SAP Fiori applications run directly in a standard web browser, making them accessible from any device and operating system. While browser-based access was technically possible before using the SAP GUI for HTML, this earlier approach relied on the traditional, function-driven navigation and lacked the responsive design and modern user experience that SAP Fiori provides. With Fiori, users benefit from a device-independent interface that adapts seamlessly to different screen sizes.

The real improvement brought by SAP Fiori lies in newly developed applications such as "Track Sales Orders." With this app, users can not only view a graphical representation of the sales document flow, but also see the sequence and status of related FI (Financial Accounting) documents created throughout the order-to-cash process. This integrated overview and visualization make it easier to monitor and analyze the order fulfillment status and financial impact of sales orders directly compared to classic SAP GUI transactions [see Figs. 7, 8].

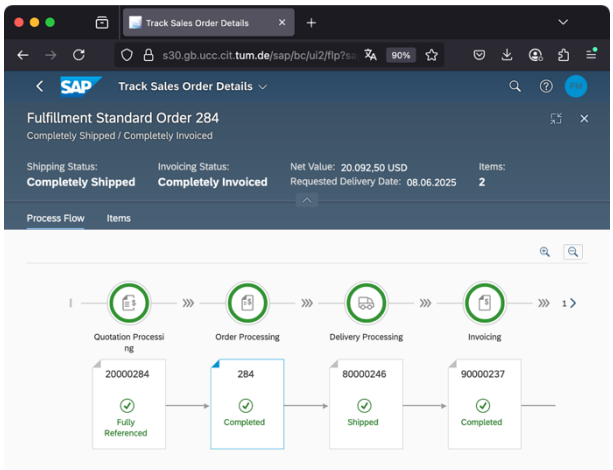


Fig. 7. Example of the SAP Fiori process flow [Screenshot from SAP S/4HANA, as of November 2025, © SAP SE. All rights reserved].

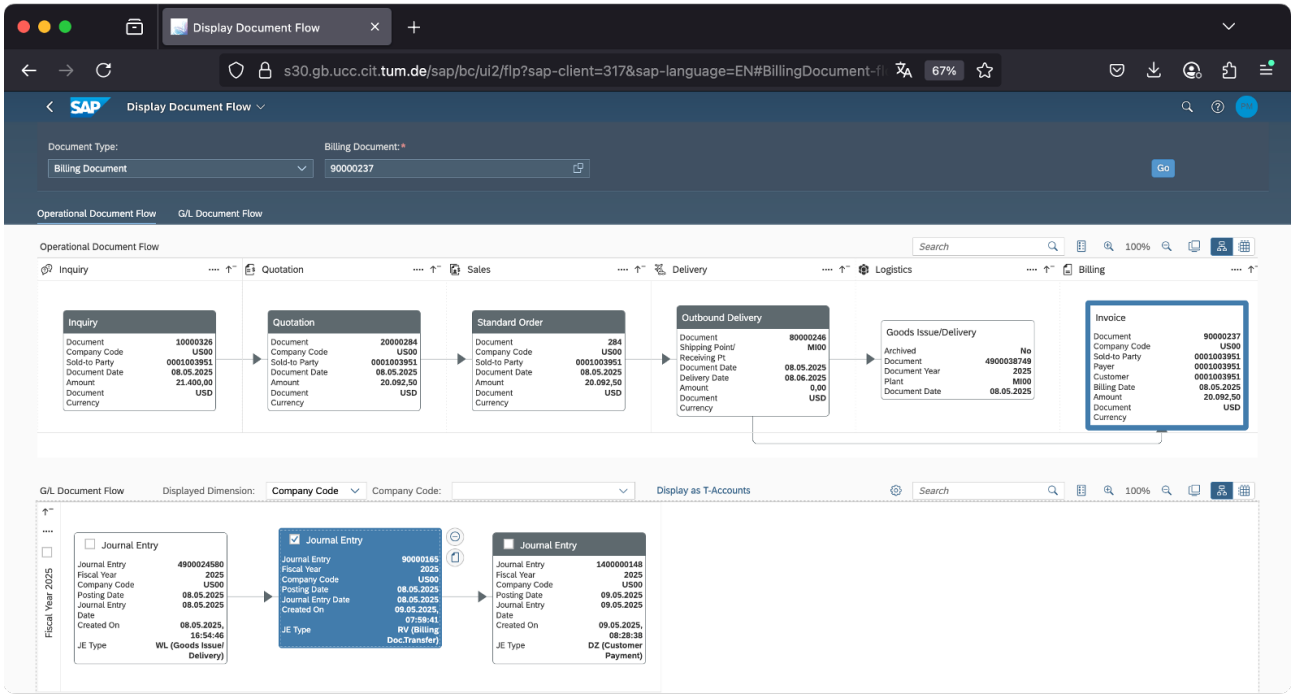


Fig. 8. Example of the SAP Fiori sales document flow with related FI documents [Screenshot from SAP S/4HANA, as of November 2025, © SAP SE. All rights reserved].

Another example is the extended error handling capabilities newly developed and provided with SAP Fiori. As shown in Fig. 9, the Sales Order Fulfillment Issues

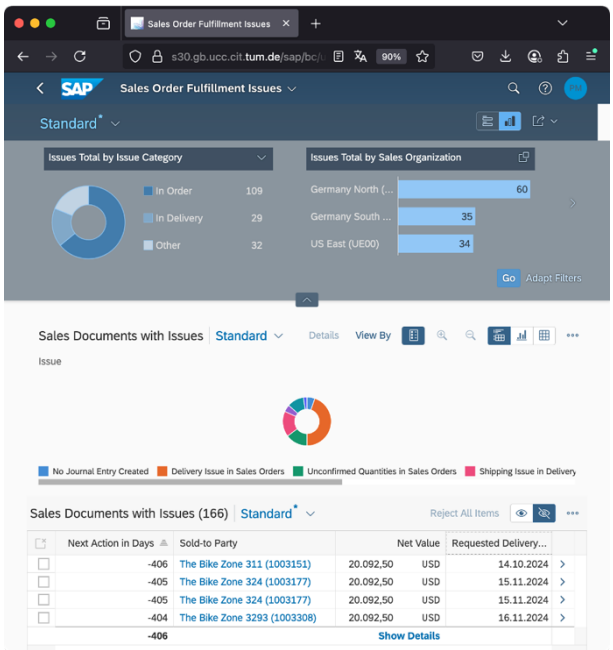


Fig. 9. Example of the SAP Fiori Sales Order Fulfillment Issues overview [Screenshot from SAP S/4HANA, as of November 2025, © SAP SE. All rights reserved].

application displays all sales documents with issues in sales orders, deliveries, and invoices – such as incomplete data or shipping problems. By selecting a specific issue, users can access detailed information, analyze the root cause, and directly correct the problem.

For a better understanding, the process flow is also visualized within the app. This visualization uses SAP Fiori's process flow control, which graphically represents the sequence and status of related documents and workflow steps [see Fig. 10]. It allows users to see the entire process path, including branching and pending steps, with interactive elements such as detailed popovers and status indicators. This helps users quickly identify where issues occur and facilitates efficient analysis and resolution directly within the application.

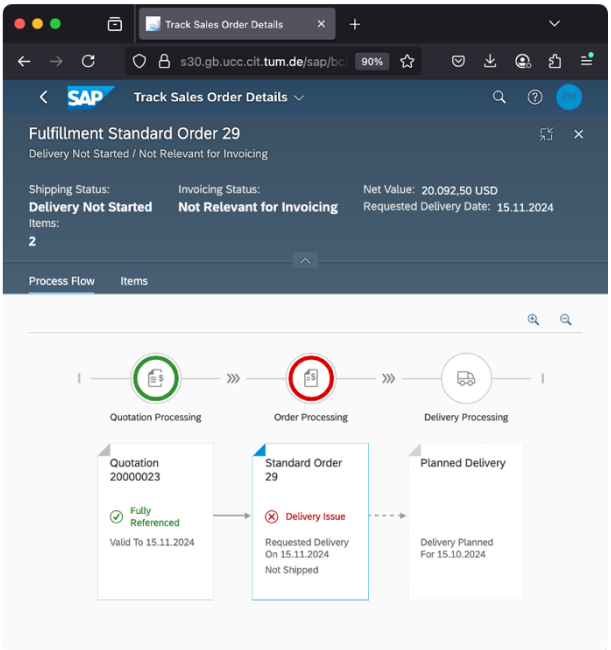


Fig. 10. Example of the SAP Fiori sales process flow and issue details [Screenshot from SAP S/4HANA, as of November 2025, © SAP SE. All rights reserved].

As a result, this application can be integrated into daily business operations even without a dedicated business monitoring concept or sophisticated process mining software. It helps save analysis time when a process gets stuck at some point in the business sequence, offering practical support for issue resolution within standard SAP Fiori capabilities.

When searching for transaction codes such as 'SPRO' or 'VA01' in SAP Fiori, the transactions appear with the familiar OK-code field in the upper left corner, allowing users to enter transaction codes just as they would in the classic SAP GUI [see Fig. 11]. At the time of publication, it remains unclear whether this is a specific characteristic of the SAP version supplied and hosted by the SAP UCC, or whether this feature is intentional and present in all currently available SAP S/4HANA versions. If it is generally available, it would seem to override some of the core logic and objectives of the SAP Fiori user experience. However, for the traditional SAP GUI user, this means that they can continue to start transactions using the transaction code as before.

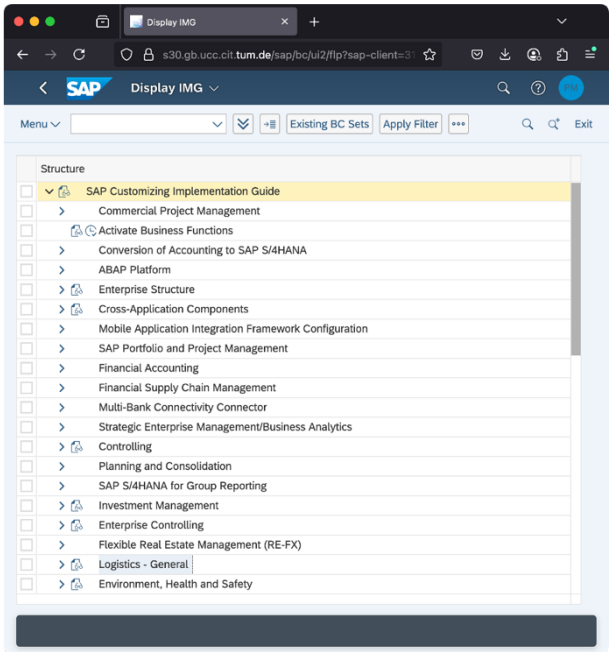


Fig. 11. Example of the SAP Fiori app for transaction "SPRO" [Screenshot from SAP S/4HANA, as of November 2025, © SAP SE. All rights reserved].

5. Summary

The brief comparison of SAP GUI and SAP Fiori highlights some key differences between the two user interfaces. SAP GUI is the traditional interface for SAP systems and has been in use since the 1990s. It offers comprehensive functionality and stable, reliable access to SAP systems, making it preferably suitable for consultants, developers, and power users who require advanced features such as customizing and ABAP development. However, its interface is often regarded as outdated and not particularly user-friendly, which can make it more challenging for new users to learn and navigate.

SAP Fiori, on the other hand, is a modern, web-based interface designed to enhance usability and user experience. It features a role-based, intuitive, and responsive access that works seamlessly across desktops, tablets, and mobile devices, making it especially suitable for business users and mobile scenarios. Fiori apps are organized as tiles on a launchpad and are tailored to specific user roles, making navigation easier and more efficient for the individual user. Fiori focuses on simplicity, personalization, and embedded analytics, and is designed for future SAP innovations.

While SAP GUI remains essential for advanced technical tasks and legacy processes, SAP Fiori significantly enhances usability, mobility, and user satisfaction – particularly for new SAP and business users looking for a more contemporary and user-friendly SAP interface.

References

[1] S. Damberg, "Über den Einsatz von SAP S/4HANA in der digitalen, internationalen Lehre," Proceedings of the SAP Academic Community Conference D-A-CH 2020, S. 2, 2020.

[2] M. Sprenger, "Schnelleinstieg in SAP Fiori," 1. Auflage, Espresso Tutorials, Gleichen, 2024.

[3] SAP SE, "SAP Fiori," SAP, 2025. [Online]. Available: sap.com/germany/products /technology-platform/fiori.html. [Accessed: May 28, 2025].

[4] A. Schaffry, "SAP S/4HANA – eine "Alles-in-die-Cloud-Strategie" ist keine Option," Computerwoche, January 6, 2025. [Online]. Available: https://www.computerwoche.de/article/3632956/sap-s-4hana-eine-alles-in-die-cloud-strategie-ist-keine-option.html. [Accessed: November 26, 2025].

[5] C. Beyer, M. Kumm, C. Lindner, "User Experience mit SAP," 1. Auflage, Rheinwerk Verlag, Bonn, 2020.

[6] L. Gloßner, "Steigerung der User Experience durch Verwendung der SAP Fiori Apps bei Schwan Cosmetics," Bachelorarbeit, Technische Hochschule Ingolstadt, 2024.

[7] Stellwerk Consulting, "SAP Fiori 3.0," Stellwerk Consulting, 2020. [Online]. Available: https://stellwerk.net/wp-content/uploads/2022/09/sap-fiori.pdf. [Accessed: May 28, 2025].

[8] S. Friedemann, S. Gröger, M. Schumann, "Was denken Studierende über SAP ERP? Ein Vorher-Nachher-Vergleich von Einflussfaktoren auf die Nutzungswahrnehmung," 5. Fachtagung Hochschuldidaktik der Informatik, 6.-7. November 2012, Universität Hamburg, Tagungsband "HDI 2012 – Informatik für eine nachhaltige Zukunft," Universitätsverlag Potsdam, S. 124, 2013.

[9] M. Engelbrecht, "SAP Fiori: Implementierung und Entwicklung," 3. aktualisierte und erweiterte Auflage, Rheinwerk Verlag, Bonn, 2020.

[10] Wikipedia, "SAP GUI," Wikipedia, May 12, 2025. [Online]. Available: https://de.wikipedia.org/wiki/SAP_GUI. [Accessed: May 28, 2025].

[11] B.-V. v. Bülow, “Loriots heile Welt,” Diogenes Verlag, Zürich, 1973.

Acknowledgements

The system used for the analysis was an SAP S/4HANA 2022, SAP Basis 757 on SAP HANA DB 2, provided by SAP University Competence Center (UCC), Garching, Germany.

All screenshots from SAP S/4HANA, as of May 2025, © SAP SE. All rights reserved.

The language editing was carried out with the support of an AI-based assistant from Perplexity.

Author Biographies

Patrick Möbert is a Professor in the Department of Computer Science and Mathematics at the University of Applied Sciences Munich. His main research interests are information systems and management, in particular SAP Enterprise Resource Planning systems, their operation, and monitoring.

Mr. Möbert studied physics in Hamburg, Germany, and Uppsala, Sweden, and graduated from the University of Hamburg in 1995. He completed his postgraduate studies in solid-state laser physics at Hamburg and Orlando, Florida, USA, earning his doctorate from the University of Hamburg in 1998.

From 1998 to 2009, he held various positions at SAP in Walldorf, Germany, as well as in the US and Japan, ultimately serving as Chief Service Architect.

Since 2009, in his role at the University of Applied Sciences Munich, Mr. Möbert has engaged in various collaborations with SAP, contributing to different divisions and areas of application.

Mr. Möbert is a member of the griffelkunst association in Hamburg and a member of the Bavarian pug society [11].

EIN KONZEPT ZUR MONETARISIERUNG UND DYNAMISCHEN ZUSCHLAGSBASIERTEN PREISERMITTLUNG VON FREIER SOFTWARE

Maximilian Overkamp
Betriebswirtschaftslehre
Universität Osnabrück
Neuer Graben 29
49074 Osnabrück
E-Mail: moverkamp@uni-osnab-rueck.de

Prof. Dr.-Ing. Andreas Schmidt
Wirtschaftsinformatik
Hochschule Osnabrück
Caprivistr. 30a
49076 Osnabrück
E-Mail: A.Schmidt@hs-osnab-rueck.de

Frank Koormann
Geschäftsführer
Intevation GmbH
Neuer Graben 17
49074 Osnabrück
E-Mail: frank.koor-mann@intevation.de

SCHLÜSSELWÖRTER

Freie Software, Commercial Open Source Software, Monetarisierung, Preisermittlung, Zuschlagsrechnung

ABSTRACT

Die Einbindung von Freier Software in die betriebliche Leistungserstellung ist mit Chancen, Herausforderungen und Spannungsfeldern verbunden. Die Monetarisierung Freier Software weist spezifische Merkmale auf, welche zur erfolgreichen Konzipierung zu beachten sind. Am Beispiel des Konferenz- und Veranstaltungssystems OpenSlides der Intevation GmbH werden die Monetarisierungskonzepte drei tertiärer Wettbewerber auf Basis des Modells von Commercial Open Source Software analysiert, um ein personalisiertes Konzept für OpenSlides als Grundlage zur Preisermittlung zu schaffen. Dieses Konzept zur Preisermittlung berücksichtigt die typischerweise nicht direkt anzurechnenden Kosten der Entwicklung von Freier Software mittels einer iterativ gestalteten Entscheidungsmatrix. Das Konzept wird in einer theoretischen Testumgebung durchlaufen, um die Eignung zur Umsetzung in die Praxis zu validieren. Die Ergebnisse werden kritisch bewertet und kontextualisiert.

EINLEITUNG

„You should think of 'free' as in 'free speech', not as in 'free beer.'" (FSF und GNU 2001) Diese Aussage der Free Software Foundation (FSF) kontextualisierte Freie Software im wirtschaftlichen Kontext. Obwohl die Ursprünge von regulierter Freier Software auf die 1980er-Jahre zurückzuführen sind, bestehen weiterhin Unklarheiten zur Implementierung von Freier Software als Teil der betrieblichen Leistungserstellung. Wie sind die Merkmale und Regeln von Freier Software mit konventionellen Methoden zur Monetarisierung von Software vereinbar? Typischerweise wird der Quellcode – die Struktur der Software – von kommerzieller Software verschlossen. Die Motivation dahinter ist es, Wettbewerbsvorteile gegenüber Konkurrenten zu erhalten und die direkte Transaktion von Geld für die Bereitstellung des Produktes an Kunden zu gewährleisten. Hier wird von proprietärer Software gesprochen. Zudem hat sich die Bereitstellung von Software aufgrund von technischen Fortschritten zunehmend in den digitalen Raum verlegt: Durch verringerten Hardware- und Personalbedarf senken Cloud-basierte Lösungen die Schwellen zur Nutzung von Software im betrieblichen Kontext. Zur Deckung des Bedarfs von neuen Kundensegmenten entstanden vollumfängliche Lösungen, welche Kunden über den Zeitpunkt der Softwarebereitstellung begleiten: Software wird nicht nur als Anwendung, sondern als Kombination von Produkten und Leistungen angeboten. Marktführer im proprietären Softwaremarkt wie Microsoft konnten durch Cloud-Software wachsende Umsatzerfolge verzeichnen. Im Jahr 2024 macht der Bereich Intelligent Cloud mit 28,5 Milliarden US-Dollar den größten Umsatzblock im Unternehmen aus (Microsoft Corporation 30.06.2024). Doch auch namenhafte Anbieter von Freien Softwarelösungen wie Red Hat konnten 2019 (vor der Übernahme durch IBM) einen Jahresumsatz von 3,4 Milliarden US-Dollar melden (Red Hat 25.03.2019). Auch Anbieter von Freier Software stellen „As-a-Service"-Lösungen zur cloudbasierten Nutzung bereit. Der Artikel baut auf Erkenntnissen zu tertiären Monetarisierungskonzepten für Freie Software auf und entwickelt daraus systematisch eine dynamische Methode zur Preisermittlung. Dazu werden klassische betriebswirtschaftliche Konzepte mit dem Stand der Technik vereint, um unter Berücksichtigung aktueller Konzepte wie dem Dynamic Pricing eine Anwendung für kommerziell verwendete Freie Software zu finden. In der Validierung werden die Erkenntnisse am Produkt OpenSlides der Intevation GmbH durchgeführt, um den Nutzen in der Praxis nachzuweisen.

KONZEPTGRUNDLAGEN

Bereitstellung von Software

Tabelle 1 zeigt eine Kategorisierungssystematik zur Aggregation von Bereitstellungsmerkmalen von Software auf, welche im weiteren Verlauf dieser Arbeit genutzt wird.

Tabelle 1: Bereitstellungsmerkmale von Software
Eigene Abbildung

<table>
<tr><td>Lizenzart</td><td>Freie Lizenz</td><td colspan="2">Proprietäre Lizenz</td><td></td></tr>
<tr><td>Laufzeit</td><td colspan="2">dauerhaft erworben</td><td colspan="2">befristet erworben</td></tr>
<tr><td>Erwerbs-modell</td><td>Einmaliger Erwerb</td><td colspan="2">Testversionen</td><td>Abonnement</td></tr>
<tr><td rowspan="2">Lizenz-umfang</td><td rowspan="2">vervielfältigbare Einzel-lizenz</td><td colspan="3">Einzellizenz</td></tr>
<tr><td colspan="3">Volumenlizenz</td></tr>
<tr><td rowspan="4">Zugriff</td><td colspan="3">Physischer Datenträger (CD, USB)</td><td rowspan="4">Cloud Computing</td></tr>
<tr><td colspan="3">Download</td></tr>
<tr><td colspan="2">quelloffen</td><td>nicht quelloffen</td></tr>
<tr><td>Hosting</td><td colspan="3">On-Premise-Hosting</td></tr>
<tr><td>Form</td><td colspan="3">On-Premise Software</td><td>SaaS</td></tr>
</table>

Da Software immateriell ist, wird mit den Erwerb der Zugang zum Produkt erworben. Neben der Software selbst wird typischerweise eine Endbenutzer-Lizenzvereinbarung zwischen dem Hersteller und dem Nutzer geschlossen. Hier bestimmt der Anbieter die Nutzungsbedingungen der Software, was maßgeblich für den vorliegende Softwaretyp ist (Brassel und Gadatsch 2019, S. 8). Die Lizenzen sind in Freie und proprietäre Arten zu kategorisieren. Der Erwerb kann eine dauerhafte oder befristet Laufzeit vorsehen. Der dauerhafte Erwerb bedeutet, dass durch eine einmalige Transaktion – in der Regel dem Kaufpreis – der permanente Zugang zur Software gewährleistet wird. Ein befristeter Zugang ermächtigt den Nutzer, die Software für ein bestimmtes Zeitintervall zu nutzen (Brassel und Gadatsch 2019, S. 8). In der Praxis manifestiert sich die zeitliche Befristung in Form eines Abonnement-Modells. Durch wiederkehrende Zahlungen wird die Befristung des Nutzungsintervalls erneuert. In der Regel werden monatliche oder jährliche Zahlungsintervalle angeboten, auch unter Erhebung einer Mindestlaufzeit (Lehmann et al. 2010, S. 161–162). Die Testversion stellt eine weitere Befristungsart dar. Hier wird die Software für eine gewisse Zeit zur Verfügung gestellt, um die Funktionsweise zu testen. Nach Ablauf des Testzeitraumes erlischt das Nutzungsrecht der Software – außer der Nutzer entscheidet sich zum Erwerb der Software. Eine andere Ausprägung ist die (un-)befristete Bereitstellung der Software mit eingeschränkter Funktionalität. Diese Erwerbsform ist Bestandteil eines Freemium-Geschäftsmodells, da der funktionale Kern kostenlos zur Verfügung steht, die Nutzung gewisser Features jedoch den kostenpflichtigen Erwerb voraussetzt (Kollmann 2018). Insbesondere Softwareanbieter im B2B-Kontext bieten Volumenlizenzen an, wodurch der Erwerb von Lizenzen zur Verwendung durch mehrere Nutzer ermöglicht wird (Microsoft Corporation 2022, S. 3). Microsoft bietet den Erwerb von Volumenlizenzen für Windows Versionen, Microsoft 365 und weitere Produkte als Teil ihres Commercial Licensing Programms an (Mittermeier 2022). Die Kundenvorteile von Volumenlizenzen sind ein vergünstigter Preis gegenüber dem Erwerb derselben Menge an Einzellizenzen sowie einem zentralisierten Lizenzmanagement. Die Laufzeit variiert: sowohl dauerhafte (Open Value) als auch befristete Lizenzen (Open Value Subscription) werden angeboten (Microsoft Corporation 2022, S. 4). Der tatsächliche Zugriff erfolgt via physischen Datenträger oder Download. Relevant ist der Umfang des Zugriffs, da die Offenheit des Quellcodes die Software-Lizenzart bestimmt.

Freie Software / Open Source Software

Eine frühe Definition im Jahr 1986 vom Gründer der FSF, Richard M. Stallman, für den Begriff „free“ (im Deutschen zur Abgrenzung großgeschrieben als Frei) lautet wie folgt: „The word ‚free‘ in our name does not refer to price; it refers to freedom.“ (Stallman 1986) Der Begriff Open Source wurde am 3. Februar 1998 von der Open Source Initiative (OSI) erstellt. Die Motivation entstand aus der Annahme, dass die Veröffentlichung des Quellcodes von Netscape die Interaktion und Teilhabe von Nutzern an der Entwicklung der Software bestärkte. Darüber hinaus wollte sich die OSI von philosophischen Konnotationen des Labels Freie Software distanzieren (Open Source Iniative 2006). Gemäß der Definition der Free Software Foundation (FSF) räumt Freie Software ihren Nutzern ein: „the freedom to run, copy, distribute, study, change and improve the software“ (FSF und GNU 2001). Die FSF Europe erkennt den Begriff Open Source als ursprünglich synonym entstanden an und räumt ein, dass Open Source weitere Kriterien umfasst als Freie Software (Free Software Foundation Europe o.J.).

Tabelle 2: Definitionsunterschiede von Freier und Open Source Software
Eigene Abbildung in Anlehnung an: (Free Software Foundation Europe o.J.; FSF und GNU 2001; Open Source Initiative 2006)

<table>
<tr><th></th><th>Verwenden</th><th>Verbreiten</th><th>Verstehen</th><th>Verbessern</th></tr>
<tr><td rowspan="2">Freie Software Definition</td><td>Zweck-ungebundene Nutzung</td><td>Kostenfreie Kopie und Weitergabe möglich</td><td rowspan="2">Untersuchen des Codes ohne Vereinba-rungen erlaubt</td><td rowspan="2">Beliebige Modifizierung und Weitergabe</td></tr>
<tr><td>Ohne Ein-schränkungen</td><td>Lizenzkosten und Tantie-men verboten</td></tr>
<tr><td rowspan="2">Open Source Software Definition</td><td rowspan="2">Ohne Dis-kriminierung von Personen, Gruppen oder Arbeitsfeldern</td><td>Lizenz darf Drittsoftware nicht einschränken</td><td rowspan="2">Offener Quelltext muss der Software beiliegen oder per Internet beziehbar sein</td><td rowspan="2">Modifika-tionen auf Wunsch des Autors in Patch-Datei statt im Quellcode</td></tr>
<tr><td>Produkt- und technologie-neutrale Lizenz</td></tr>
</table>

Aufgrund der a) hohen inhaltlichen Überschneidung und b) geringen Relevanz der Unterschiede zur Konzipierung des Monetarisierungskonzeptes wird in diesem Artikel nicht zwischen Freier und Open Source Software differenziert – im Weiteren wird von Freier Software gesprochen.

Proprietäre Software

Proprietäre Software sind Anwendungen mit Lizenzen, die ihren Nutzern die zuvor erwähnten Freiheiten nicht einräumen. Ergo schränkt proprietäre Software die Nutzung, Einsicht, Weitergabe und Veränderbarkeit durch den Nutzer ein, beispielsweise durch die Verschließung des Quellcodes. Die FSF definiert wie folgt: „Putting some of the freedoms off limits to some users, or requiring that users pay, in money or in kind, to exercise them, is tantamount to not granting the freedoms in question, and thus renders the program nonfree" (FSF und GNU 2001). Es ist anzumerken, dass Freie Software nicht pauschal im Zielkonflikt mit der betrieblichen Ertragsgeneration stehen. Sofern die erläuterten Merkmale nicht eingeschränkt werden, handelt es sich weiterhin um Freie Software.

Cloud Computing

Die Zugriffsmethoden sind in zwei Orte zu unterteilen: Cloud und On-Premises. On-Premise Software sind Anwendungen, die nur lokal verfügbar sind und zur Benutzung keine Verbindung zur Cloud voraussetzen. Dieser Zugriffsort umfasst eine private Cloud, da der Softwareanbieter nicht in der Bereitstellung involviert ist. Die Cloud-Bereitstellung über den Hersteller bedeutet, dass die Verwendung nur über die Cloud verfügbar ist und steht in Verbindung mit diversen Servicemodellen (Mell und Grance 2011). Der Cloud-Zugriff involviert den Anbieter über den Erwerbszeitpunkt der Software hinaus in die Bereitstellung und kann näher in den Begriff Cloud Computing präzisiert werden. Cloud Computing beschreibt die cloudbasierte Bereitstellung von IT-Dienstleistungen und Infrastruktur im Zusatz zur eigentlichen Anwendung. Der Zugriff auf die Software und das Hosting erfolgt rein online und wird durch den Anbieter, in der Regel im Rahmen eines Abonnements übernommen (Sehgal und Bhatt 2018, S. 2). Durch die Übertragung der technischen Konfiguration und Betrieb der Software in den Verantwortungsbereich des Anbieters wird für den Kunden spezifisch der Zugriff auf ihre Softwareumgebung gewährleistet (Cheng 2024, S. 37). Die Betrachtung ist für diesen Beitrag von besonderer Relevanz: Im Jahr 2023 hat das Statistische Bundesamt erhoben, dass 33% der selbstständig wirtschaftlich tätigen Einheiten in Deutschland kostenpflichtige IT-Dienste per Cloud Computing beziehen. Davon beziehen Unternehmen mit mehr als 250 Angestellten (78%) sowie mit 50 bis 249 Angestellten (59%) im Vergleich zu kleinen Unternehmen (1 bis 9 Angestellte: 31%; 10 bis 49 Angestellte: 43%) häufiger Cloud Computing Dienste. Zu den Hauptverwendungszwecken gehören E-Mail Services sowie Anwendungen in den Bereichen Office, Finanz- oder Rechnungswesen. Die Speicherung von Daten wird von insgesamt 70% aller Einheiten als Cloud Service bezogen (Statistisches Bundesamt 2023). Zeitraumbezogen zeigt eine Studie, dass sich in den Jahren 2011 bis 2022 der Anteil der nutzenden Unternehmen von Cloud Computing invers zum Anteil der Planer und Diskutierter entwickelte. (KPMG und Bitkom Research 2022) Vorhaben via Cloud Computing werden häufiger umgesetzt: das Vertrauen in die Technik steigt.

Software as a Service

SaaS ist eine Form der Bereitstellung von Software an ihre Nutzer. Im Kontext der Bereitstellungsmethoden von Software wird das Produkt nicht auf dem Endgerät des Nutzers installiert, sondern online abgerufen (Cheng 2024, S. 147). Dieses Portal zur Interaktion mit der Software wird vom Hersteller in Form eines Web-Interfaces oder einer Cloud-basierten Anwendung bereitgestellt. Alle Abläufe in der Anwendung erfordern neben der Verbindung mit dem Internet den Zugriff auf einen Anwendungsserver sowie auf einen Datenbankserver (Cheng 2024, S. 148). Diese Server werden zum Abruf der Anwendung inklusive eingetragener Daten verwendet. Die Software befindet sich im eigentlichen Sinne nicht im Besitz des Kunden. Via Lizenzvertrag wird lediglich das Recht zur Interaktion mit der Anwendung im spezifizierten Rahmen gewährleistet. Die Software auf technischer Ebene befindet sich im Verfügungsbereich des Anbieters. Hieraus ergibt sich der zentrale Leistungsaspekt von SaaS: Der Kunde kauft neben der Software auch die erforderliche Infrastruktur als Leistung zum Betrieb der Software für den spezifizierten Zeitraum. Im Falle vom Cloud Computing stellt der Anbieter die erforderliche Hardware zum Betrieb der Software in Form von Serverkapazitäten bereit (Cheng 2024, S. 148). Abseits von gesetzlichen Gewährleistungsansprüchen sind die Pflege und Wartung von Software nicht zwingend Produktbestandteil. Nach dem Erwerb der Software obliegt es grundsätzlich dem Kunden, die Software einzurichten und mithilfe der notwendigen Hardware zu betreiben. Im privaten Kontext involviert dies typischerweise nur den Nutzer und das Endgerät. Die Anwendung im Unternehmen, wo die kollaborative Arbeit den Zugriff mehrerer Endnutzer auf eine Software oder Datenbank erfordert, kompliziert den Anwendungsfall. Hieraus ergeben sich mehrere Geschäftsmodelle, welche die Bereitstellung der Software mit spezifischen Dienstleistungen verknüpfen. Im Falle von SaaS übernimmt der Anbieter auch die Leistungen, welche nach dem Erwerbszeitpunkt und der Bereitstellung der Software anfallen. Dazu gehört das Onboarding, die Implementierung der neuen Anwendung in die bestehende IT-Systemlandschaft des Kunden, um die Inbetriebnahme zu gewährleisten. Fehlerbehebungen, Wartungen, sowie die Softwareaktualisierung im Falle von neuen Updates gehören als fortlaufende Leistungen dazu. Mitarbeiterschulungen zum Einsatz der Software können ebenfalls Teil des Leistungspaketes sein (Cheng 2024, S. 148). Somit ist bei SaaS die Software als eigent-

liches Produkt der Kern, die ergänzenden Leistungsebenen bilden das gesamte Produkt zur Bedienung von Kundenbedürfnissen.

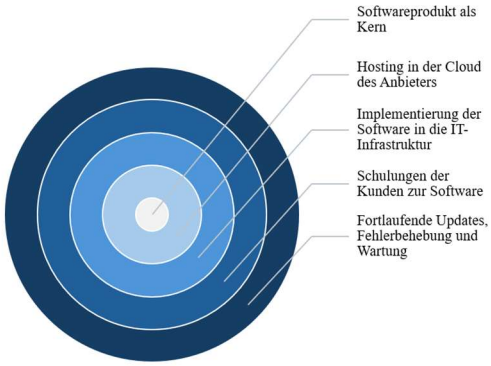


Abbildung 1: Produkt-Leistungsumfang von SaaS
Eigene Abbildung in Anlehnung an: (Cheng 2024, S. 147–151)

Commercial Open Source Software

Um Freie Software nach dem Verwendungszweck zu unterscheiden, wurde für die betriebliche Verwendung von Freier Software zur Gewinnerzielung der Begriff „Commercial Open Source Software“ (COSS) als Abgrenzung zur „Community Open Source Software“ (OSS) gebildet (Riehle 2007, S. 25–32). Obwohl dieses Konzept von „Open Source” spricht, ist es aufgrund der zuvor behandelten Unterschiede zwischen Freier und Open Source Software im Kontext von Freier Software anwendbar. Im Folgenden wird COSS im Blinkwinkel von Freier Software untersucht und zur Erstellung eines Monetarisierungskonzeptes angewandt. COSS bezieht sich auf eine Anwendung, welche durch Akteure wie ein Unternehmen oder eine Einzelperson mit wirtschaftlichem Interesse entwickelt wird. Sie besitzen die Software, im Sinne, dass das Urheberrecht bei Ihnen liegt. OSS bezieht sich hingegen auf Software mit einem gemeinschaftlichen Interesse und Entwicklung, oftmals auf freiwilliger Basis (Riehle 2007, S. 25). Shahrivar et al. haben auf Basis der Erkenntnisse über COSS-Geschäftsmodelle folgende Bestandteile für Freie Software und monetäre Komplemente (MK) definiert:

Tabelle 3: Das COSS-Modell
Eigene Abbildung in Anlehnung an: (Shahrivar et al. 2018, S. 205)

Kategorien	Beschreibung
Freie Software als Kernprodukt	Die eigentliche und unabhängig funktionierende Anwendung, welche regulär als Freie Software zur Verfügung steht.
Monetäre Softwarekomplemente	Ergänzende Features zum Softwarekern, die unabhängig vom Freien Kern bereitgestellt und lizensiert werden können.
Monetäre Servicekomplemente	Allgemeine und spezifische Dienstleistungen zur Software, die nach Kundenwunsch/Leistungsspektrum umgesetzt werden.
Monetäre Hardwarekomplemente	Güter und Software mit eingebetteten Systemen zur Erweiterung der Funktionalität des Kernprodukts im Anwendungskontext

Die Ausgestaltung der monetären Softwarekomplemente variiert in der Praxis. Zum einen können die Features der Freien Software in Form von optionalen Erweiterungskomplementen ergänzt und spezifiziert werden. Manche IT-Unternehmen für Freie Software, wie die Intevation GmbH, bieten die zielgerichtete Entwicklung von Features auf Kundenwunsch an, um die Software für den Use-Case des Kunden vorzubereiten. Zum anderen kann darunter eine Freistellung eines Anbieters verstanden werden, die monetären Softwarekomplemente unter einer neuen Version zu bündeln und diese unter einer abweichenden Lizenz zu veröffentlichen. Dies umfasst Dual Licensing, in welchem Versionen derselben Software unter verschiedenen Softwarelizenzen veröffentlicht werden (Comino und Manenti 2011, S. 235).

FORSCHUNGSERGEBNISSE

Forschungsmethode

Zur Erstellung eines geeigneten Monetarisierungskonzeptes für OpenSlides wurden im Rahmen dieser Arbeit drei tertiäre Wettbewerber im Bereich von Freier Software analysiert: Das Datenbankmanagementsystem MySQL, das On-Premise-CRM SugarCRM und das WordPress-Plugin WooCommerce. Es wurde die Monetarisierung, das Produkt-Ökosystem und die Unternehmensgeschichte untersucht. Im Folgenden werden relevante Erkenntnisse, die Auswahl eines Konzeptes und das daraus zugeschnittene Monetarisierungskonzept für OpenSlides vorgestellt. Diese bilden die Grundlage für das anschließende Konzept zur Preisermittlung von Freier Software und ihren MK.

Vorstellung der Analyseergebnisse

MySQL ist eine Anwendung zur Erstellung und Verwaltung von relationalen Datenbanken im lokalen oder webbasierten Kontext (Oracle o. J.a). MySQL ist in fünf Produkteditionen aufgeteilt: Community, Classic, Standard, Enterprise und Cluster CGE. Die Community Edition ist die Freie Softwarevariante von MySQL und steht kostenlos unter der GPLv2 zum Download verfügbar. (Oracle 2024a) Die weiteren Versionen unterstehen einer proprietären Lizenz. (Oracle o. J.b) Für MySQL wird ein duales Lizenzmodell angewandt: Die Community Edition ist Frei unter der GPL, die Editionen Standard, Enterprise und Cluster CGE proprietär.
Die proprietären Editionen sind als SaaS-Abonnement in Verbindung mit dem Hosting auf Oracle-Servern verfügbar. (Oracle o. J.d) Standard, Enterprise und Cluster CGE sind für Endnutzer im Privat- und Unternehmensbereich konzipiert und unterscheiden sich im Produkt-Leistungsumfang. Anwendungsspezifische Softwarekomplemente wie der NDB Cluster Manager sind den entsprechenden kommerziellen Versionen vorbehalten. Die Buchung einzelner Features ist nicht möglich, womit auf Kundenseite der Wechsel in eine andere Version zur Erweiterung des

Produktumfangs notwendig ist. Die kommerziellen Editionen werden als jährliches Abonnement im SaaS-Modell angeboten. Als Servicekomplement wird der Oracle Lifetime Support for MySQL angeboten. Das Abonnement bezieht sich auf den Produktlebenszyklus und verlängert die Laufzeit für Unterstützungsdienstleistungen von Oracle und ist in die Stufen Premier (Jahre 1-5), Extended (Jahre 6-8) und Sustain (ab 9 Jahre) aufgeteilt. Für MySQL Abonnementkunden ist der Premier Support inbegriffen, Einmalkauf-Lizenznehmer und Nutzer der Community Edition können diese separat erwerben. Der Premier Support beinhaltet die Wartung und Updates sowie allgemeine Fehlerbehebungen. Extended Support verlängert die Fehlerbehebung um 3 Jahre und beschränkt den Geltungsbereich aller Dienstleistungen auf spezifische MySQL-Releases. Sustaining Support ermöglicht den Zugriff nur auf existierende Updates oder Fehlerbehebungen. (Oracle o. J.c) Für die Community Edition werden keine Updates bereitgestellt, somit beschränkt sich der Service auf Fehlerbehebungen und den allgemeinen Kundendienst (Oracle 2024b, S. 12). Aus der dualen Lizensierung ergeben sich mehrere Vorteile: Durch die kostenfreie Community Edition wird die Verbreitung des Datensystems intensiviert, da die Adaptionskosten insbesondere für neue Unternehmen niedrig sind. Die proprietären Editionen generieren unmittelbaren Umsatz aus skalierbaren Abonnements und zusätzlichen Supportleistungen. Da private Unternehmen nicht zwingend Freie Software priorisieren oder Auflagen zur Nutzung haben, agiert die Community Edition mit wachsender Arbeitslast als Freemium-Version, bis ein erweiterter Funktionsumfang oder zusätzliche Leistungen benötigt werden. In Bezug auf OpenSlides ergeben sich Problematiken: Durch die Veränderbarkeit des Freien Quellcodes durch Nutzer und im Hinblick auf die Skalierung der Reichweite von OpenSlides und Komplexität der Anwendung werden präzise Supportleistungen erschwert. Durch die Berechnung der Servicekosten durch die Unterstellung entgangener Abonnementgebühren wie Oracle könnten hohe Barrieren für die Inanspruchnahme aufgebaut werde, da im Gegensatz zu proprietären Anwendungen die Kunden nicht abhängig von Serviceleistungen des Herstellers sind. Zudem ist die Position von MySQL Community als Marktdurchdringungsmethode abgeleitet auf OpenSlides kritisch zu sehen: Inwiefern ist der Skalierungsbedarf vorhanden, dass Kunden tatsächlich auf eine leistungsfähigere Variante des Konferenzsystems wechseln?

SugarCRM

SugarCRM ist eine Anwendung zum Kundenbeziehungsmanagement, welche Aufgaben in den Bereichen Marketing, Vertrieb und Service ermöglicht oder mit künstlicher Intelligenz automatisiert (SugarCRM o. J.). SugarCRM wurde 2004 als Freie Software veröffentlicht und konnte innerhalb von 10 Jahren weltweit verbreitet werden und eine Nutzerbasis mit Schätzungen von 1,4 Millionen aufbauen. Im Jahr 2011 wurde SugarCRM 6.2 in den Editionen Community, Professional und Enterprise angeboten. Die Editionen sind dual lizensiert, mit der GPLv3-lizensierten Community Edition und den proprietären Professional und Enterprise Editionen. Ein Vergleich zeigt, dass die proprietären Editionen exklusive Softwarekomplemente beinhalten. 7 von 19 Funktionen sind Bestandteil der Community Edition, die verbleibenden 12 setzen ein Upgrade voraus (SugarCRM 2011a). Servicekomplemente wie Kundensupport, die Bereitstellung als SaaS oder On-Site-Hosting sind den Editionen Professional und Enterprise vorbehalten (SugarCRM 2011b). 2014 wurde das Ende der Freien Softwarevariante angekündigt, da SugarCRM einen Interessenkonflikt zwischen der Beibehaltung von Freier Software und der effizientesten Bereitstellung von CRM-Software sahen. Ebenso wurde in Befragungen von Nutzern der Sugar Community Edition festgestellt, dass deren Kundenbedürfnisse (Personalisierbarkeit und Kosteneinsparungen) nicht mit der Community Edition bedient werden können (Oram 2014). In der Pressemeldung im Jahr 2017 wurde die fortlaufende Unterstützung der Community Edition im Rahmen einer Version 6.5 angekündigt, jedoch wurde das Freie Softwareprojekt im Jahr 2018 vollständig beendet (Oram 2018). Der strategische Paradigmenwechsel zu einer komplett proprietären Anwendung wirft Fragen über den Erfolg auf, da dieses Vorhaben einen Extrempunkt in der Handhabung von Freier Software darstellt. Die Einstellung der Community Edition resultierte in Forks von SugarCRM wie SuiteCRM, welche das CRM-System als separates Projekt weiterentwickeln (SuiteCRM 2017). Die Abkehr von Freier Software hatte zur direkten Folge, dass das eigene GPL-lizensierte (Diedrich 2007) Produkt als Nährboden von Konkurrenten diente, um Nutzer der Community Edition anzusprechen. Folglich wurde der Anreiz zur Umstellung auf die proprietäre SugarCRM-Variante in diesem Segment verringert. Zwar besteht im Kundensegment der Öffentlichen Verwaltung ein Teilsegment, welches Freie Software ablehnt (Schnaak und Termer 2023, S. 48). Diese 23% können durch die Abkehr von Freier Software erreicht werden, jedoch stehen die Inklusion oder ein kompletter Strategiewechsel zu proprietären Lizenzmodellen im unmittelbaren Konflikt mit der Identität als IT-Unternehmen für und mit Freier Software. Zudem besteht das Risiko, das größere Teilsegment, welches Freie Software gegenüber aufgeschlossen ist (40%) als Opportunitätskosten für die Neuausrichtung in Kauf zu nehmen (Schnaak und Termer 2023, S. 48). Die Aufgabe der Position als Spezialist für Freie Software würde die strategische Position der Intevation GmbH verändern und eine klare Positionierung zur proprietären Software verlangen, um Wettbewerbsvorteile zu generieren (Porter 1991, S. 102). Die Problematik ist, dass sich die Marktbedingungen und Konkurrenz verschieben: Als Freies System für Konferenzen und Abstimmungen besetzt OpenSlides eine Nische, welche vollumfänglich durch wenige Unternehmen bedient wird. Als direkte Konkurrenz für die Teilfunktion Videokonferenzen bestehen Lösungen wie Jitsi und BigBlueButton. Der Vorteil an Freier Software ist hier, dass diese als Bestandteil der eigenen Lösung integriert werden können, um Entwicklungskompetenzen

und -aufwände auszulagern und effizienter den Funktionsumfang des eigenen Produkts zu erweitern (OpenSlides-Team o. J.c). Die Intevation GmbH als vertikal integriertes Unternehmen für Freie Software erschafft Wettbewerbsvorteile durch vollumfängliche Lösungen. Zudem sind neue Entwicklungen und Strategien durch den öffentlich zugänglichen Code einsehbar und Maßnahmen können proaktiv vorgenommen werden. Die Marktsituation für proprietäre Lösungen unterscheidet sich, da Zoom als Marktführer (71,67%) nur geringe Marktanteile für Konkurrenten zulässt (Datanyze 2024). Die Marktführerschaft ist nicht zwangsläufig eine realistische oder erstrebenswerte Position für Unternehmen, jedoch bedeutet diese Ausgangslage, dass die Intevation GmbH neue Alleinstellungsmerkmale finden müsste, um die Position des Spezialisten zu erlangen und ihre Produkte zu monetarisieren (Porter 1991, S. 102).

WooCommerce

WooCommerce ist ein Plugin für WordPress, eine Freie Software zum Erstellen eigener Websites. Das Plugin integriert eine E-Commerce Plattform, womit Nutzer ihre WordPress-Website um einen Online-Shop erweitern (Woo o. J.e). Trotz der direkten Verknüpfung zu WordPress ist WooCommerce ein eigenes Produkt, da es maßgeblich für die spezifische Nutzung als E-Commerce Plattform ist und selbstständige Methoden zur Monetarisierung aufweist. Die Monetarisierung basiert auf zwei Modellen: Woo Enterprise und WooCommerce Marketplace. Woo Enterprise ist ein Paket an Servicekomplementen, geleistet durch Woo oder Partner des Unternehmens. Woo unterstützt Kunden bei der Migration des E-Commerce von einer Drittanwendung zu WooCommerce. Dazu gehören die Datenmigration und Konfiguration der WordPress-Seite für das Plugin. Enterprise-Kunden werden im allgemeinen Kundensupport priorisiert und erhalten individuelle Unterstützung im Onboarding. Die Kosten für Woo Enterprise werden auf Basis der Unternehmensgröße des Kunden berechnet (Woo o. J.a). Das Hosting lagert Woo auf ausgewählte Drittunternehmen und die Schwestergesellschaft WordPress aus (Woo o. J.d). Das Outsourcing des Hostings birgt strategische Vorteile für Woo: Als Schwestergesellschaften kann die Auslagerung auf WordPress.com als Teil der Strategie von Automattic Inc. zur Fokussierung auf die Kernkompetenzen gesehen werden. WooCommerce ist als Plugin auf technischer Ebene mit WordPress verflochten. Die Beteiligung Dritter am Unternehmensprozess ist ausschlaggebend für Freie Software sowie das Angebot von monetären Komplementen von COSS durch das entwickelnde Unternehmen oder Wettbewerber. Die Regulierung erfolgt durch Woo auf zwei Ebenen. Consulting-, Marketing- und Entwicklungsleistungen sowie Servicekomplemente, sind ebenfalls Teil des Partnerprogrammes. Woo Agency Partner sind externe Unternehmen mit Kompetenzen in WordPress oder WooCommerce (Woo o. J.g). WooCommerce Marketplace ist eine Plattform für Woo und Partner zum Verkauf von Software- und Servicekomplementen (Woo o. J.c). Händler erhalten 70% des Nettoumsatzes für nicht-exklusive Produkte, Woo behält sich die verbleibenden 30% als Provision vor. Die Komplemente dürfen nur zeitlich nutzungsabhängig als jährliches oder monatliches Abonnement monetarisiert werden (Woo o. J.b). Extensions ermöglichen Schnittstellen oder spezifische Funktionen. Am Beispiel einer Extension von Woo wird das Generieren von Produktempfehlungen beim Kaufprozess angeboten. Kundensupport zur Installation, Konfiguration und Nutzung sowie Updates sind im Preis gebündelt. Mit über 20.000 aktiven Installationen und einem jährlichen Abonnementpreis von $99 kann für diese Extension auf ein Umsatzvolumen von mindestens $1.980.000 pro Jahr geschlossen werden (Woo 2019). Zudem werden ergänzende Extensions in Produkt-Paketen nach Anwendungsfällen gebündelt verkauft (Woo o. J.i). Die Preishöhe ist additiv, es entstehen keine Preisvorteile für den Kunden (Lehmann und Buxmann 2009, S. 520). Der Marktplatz stellt Extensions und Themes im Abonnement zur Spezifizierung des Funktionsumfangs und Personalisierung bereit. Als Dritter muss dem Partnerprogramm beigetreten werden, um eigene Produkte anzubieten. Für nicht-exklusive Produkte fordert Woo für jeden Verkauf eine prozentuale Provision. Diese Maßnahme birgt einen Lenkungszweck, da der erzielte Umsatz für Woo-exklusive Anbieter steigt. Zudem minimiert Woo durch Exklusivpartnerschaften das Risiko von konkurrierenden Forks. Aus dem Marketplace ergeben sich beidseitige Vorteile: Kunden können ihre Software gegen eine fest kalkulierbare Summe auf ihren Anwendungsfall personalisieren und Woo generiert wiederkehrende Zahlungsströme aus den eigenen Extensions oder Provisionen der Angebote von Partnern. Im Gegensatz zu einmaligen, leistungsbezogenen Zahlungen aus klassischen Entwicklungsaufträgen bieten sich mehrere Vorteile: Wiederkehrende Zahlungen aus standardisierten Softwarekomplementen lassen sich schnell in den Softwarekern integrieren und bieten eine feste finanzielle Planungsgrundlage für die Laufzeit des Abonnements. Auf Kundenseite ist der Verzicht auf das Abonnement von Extensions mit Opportunitätskosten verbunden. Neben den Funktionen des Softwarekomplements fallen auch die beinhalteten Servicekomplemente wie Kundensupport und Updates weg – der Kunde muss auf die eigenen Kompetenzen oder Drittanbieter zurückgreifen, was mit Umstellungskosten und Effizienzverlusten verbunden ist. Unternehmen außerhalb des IT-Sektors nehmen den Wechsel von Software oder Anbietern als riskant wahr, da der Umfang der Kosten und die Übereinstimmung von Bedürfnissen und Kompetenzen in der langfristigen Zusammenarbeit entscheidend sind (Urban & Vogel 2015, S. 157). Durch die Kombination an Software- und Servicekomplementen kann der wahrgenommen Fachkräftemangel in der öffentlichen Verwaltung erleichtert werden, da hier fortwährender Support durch den Hersteller für die Abonnementdauer inbegriffen ist (Schnaak und Termer 2023, S. 50). Zudem sind Öffentliche Verwaltungen, die keine pauschalen Vorteile im Einsatz Freier Software sehen (16%), durch spezifische Erweiterungen zur Vereinfachung des individuellen Anwendungsfalls an OpenSlides

zu binden (Schnaak und Termer 2023, S. 49). Bislang wurden standardisierte Softwarekomplemente als Teil des Produkt-Ökosystems nicht erschlossen. Dies bietet Potenziale zur Steigerung des Umsatzes durch die Kombination bestehender Monetarisierungsmethoden im Bereich der Servicekomplemente mit der Monetarisierung von Softwarekomplementen.

Nutzwertanalyse

Die Nutzwertanalyse ist ein Instrument zur Bewertung mehrerer Alternativen anhand von gewichteten Kriterien, welche zur Auswahl einer Option anhand von rationalen Kriterien verwendet wird (Woock et al. 2022, S. 16). Im Rahmen einer qualitativen Befragung wurden folgende Kriterien für die Intevation GmbH zur Auswahl eines Monetarisierungskonzeptes erarbeitet:

1. Vereinbarkeit mit der Unternehmensphilosophie
2. Kompatibilität mit der aktuellen Monetarisierungsmethode
3. Entgegenwirken der Herausforderungen von OpenSlides
4. Übereinstimmung mit den Interessen des aktuellen Kundensegmentes von OpenSlides
5. Kompatibilität der Monetarisierungsmethode mit den Strategien der Intevation GmbH zur Skalierung von OpenSlides

Als IT-Dienstleistungsunternehmen für Freie Software ist diese zur betrieblichen Leistungserstellung zentral. Die Vereinbarkeit von Freier Software mit dem neuen Monetarisierungskonzept ist daher der Bestandteil mit dem größten Gewicht (35%), da die Unternehmensphilosophie zentraler Bestandteil der besetzten Marktnische als Spezialist für Freie Software ist. Die Nachteile der Abkehr von dieser Position wurden in der Analyse von SugarCRM festgestellt. Die Kompatibilität mit der aktuellen Monetarisierungsmethode fließt mit nächsthöchster Gewichtung (20%) ein, da ungeprüfte Modifikationen das Risiko aufweisen, aufgrund von Anlaufkosten ohne ursprünglichen Umsatzstrom zu Einbußen oder Verlusten zu führen. Zu den Herausforderungen von OpenSlides gehören zwei zentrale Aspekte: Entgangene Umsätze durch Selbsthoster und Nachfragefluktuationen. Selbsthoster resultieren aus der Quelloffenheit von Freier Software, durch welche OpenSlides autark, ohne die Inanspruchnahme der (Hosting-) Dienstleistungen genutzt wird. Nachfragefluktuationen resultieren aus externen Faktoren, da Konferenzsoftware pandemiebedingt einen starken Nachfrageanstieg verzeichnen konnte. Der Erhalt des Nachfrageniveaus ist anzustreben, weshalb die Generation neuen Kundennutzens notwendig ist. Dazu ist auch die Berücksichtigung der Interessen des anvisierten Kundensegments von OpenSlides, Öffentlicher Verwaltungen und verwandter Organisationen, von konkreter Relevanz. In der Besprechung der Unternehmensstrategie für OpenSlides sind zwei Aspekte von besonderer Relevanz: Personalisierbarkeit und Skalierbarkeit. Konzepte, welche die aktuelle Strategie komplementieren, sind eine höhere Effizienz in der Umsetzung zuzuschreiben und daher bevorzugt auszuwählen. Die Bewertung der Kriterien erfolgte auf einer Skala von 1 (niedrige Übereinstimmung) bis 3 (hohe Übereinstimmung).

Tabelle 4: Nutzwertanalyse
Eigene Abbildung

K	Gew.	MySQL		SugarCRM		WooComm.	
1	35%	2	0,7	1	0,35	3	1,05
2	20%	1	0,2	1	0,2	2	0,4
3	20%	3	0,6	2	0,4	3	0,6
4	10%	2	0,2	1	0,1	3	0,3
5	15%	2	0,3	2	0,3	3	0,45
Gew. Score		2		1,35		2,8	
Priorität		2		3		1	

Auf Basis der Nutzwertanalyse wurde WooCommerce als geeignete Grundlage zur Erstellung eines neuen Monetarisierungskonzeptes ausgewählt.

Monetarisierungskonzept „OpenSlides Plus“

Das Monetarisierungskonzept „OpenSlides Plus“ basiert auf dem modularen Angebot von monetären Softwarekomplementen für OpenSlides. Um unberechenbare Faktoren zu entfernen und den organisatorischen Aufwand zu verringern, ist das Konzept zunächst ohne Beteiligung von Drittanbietern geplant. OpenSlides wird als Freies Kernprodukt unter der MIT-Lizenz beibehalten, da die Lizenzänderung für das Teilsegment mit Unsicherheiten gegenüber der Lizensierung von Freier Software abschreckend wirken und negative Effekte verursachen kann (Schnaak und Termer 2023, S. 49). Der Verkauf von Freien Erweiterungen ist unter Berücksichtigung der vier Freiheiten grundsätzlich zulässig (FSF und GNU 2001). Somit sollte der Quellcode von Erweiterungen nicht pauschal wie der Freie Softwarekern auf GitHub öffentlich einsehbar sein, sondern exklusiv für Abonnementen. Somit wird gewährleistet, dass Zahlungsströme aus dem Softwarekomplement entstehen. Trotzdem besteht das Risiko, dass Dritte den Quellcode weitergeben oder öffentlich zur Verfügung stellen. Hier greifen zwei Mechanismen: Zum einen wird über das Softwarekomplement hinaus Kundennutzen durch die Servicekomplemente geschaffen: Fehler bei der Implementierung der Erweiterung ohne Abonnement liegen nicht im Zuständigkeitsbereich der Intevation GmbH. Zum anderen müssten auch alle Updates durch Dritte weitergegeben werden. Hier ist auf ein zuvor erwähntes Gentlemen Agreement zu vertrauen, welche den Aufwand durch die ständige Weitergabe mit dem erzielten Mehrwert für den Weitergeber ins Verhältnis stellt, da dieser für den Zugang zahlt (Mecke 2018). Diese Praktik wendet auch Woo an: Die GPLv3-lizensierten Extensions sind nach Abonnement in einer .zip-Datei über den WooCommerce-Account herunterzuladen

(Woo o.J.f). Updates für OpenSlides werden innerhalb von 2 bis 4 Wochen veröffentlicht und neue Features auf Anregung oder Auftrag von Kunden entwickelt (Intevation GmbH 2024). Diese Ausgangslage wird als Basis zur Erstellung eines Sortiments für Softwarekomplemente, im Folgenden als Erweiterungen bezeichnet, verwendet. Im Allgemeinen sind Erweiterungen zu veröffentlichen, die als neue Funktionen für OpenSlides in der Pipeline existieren. Somit kann der Planungsanteil der Entwicklungszeit verringert werden und ein Sortiment aufgebaut werden, welches verschiedene Bedürfnisse bedient. Zur Steigerung der Attraktivität des Konzepts wurde auf Basis der Strategien für OpenSlides ein Konzept entwickelt: In der persönlichen Kommunikation mit der Intevation GmbH wurde Crypto-Voting für OpenSlides als Teil der kurzfristigen Strategie zur Erhöhung der Sicherheit von Abstimmungen besprochen (Intevation GmbH 2024). Die Einbindung einer neuen Wahlmethode in das Konzept bietet entscheidende Vorteile in der Steigerung der Attraktivität von Erweiterungen aus Kundensicht. Die Rechtssicherheit des aktuellen Verfahrens zur elektronischen Stimmabgabe wurde durch externe Gutachten festgestellt (OpenSlides-Team o. J.a). Ein verschlüsseltes Wahlverfahren, beispielsweise auf Basis von kryptographischen Methoden oder Blockchain, würde die Geheimhaltung der individuellen Abstimmungsergebnisse erhöhen. Am Beispiel von Blockchain-Wahlverfahren können selbst Administratoren während der Abstimmung in der Datenbank nur sehen, ob eine Person abgestimmt hat. Die Abstimmungsergebnisse bleiben verschlüsselt (Stanciu et al. 2023, S. 3). Zur Gewährleistung eines erhöhten Sicherheitsstandards ist die Beteiligung der Intevation GmbH als unabhängige und durchführende Partei notwendig. Somit sind Kunden an die Inhalte und Leistungen des Abonnements gebunden, um eine verschlüsselte Abstimmung zu gewährleisten. Das ursprüngliche Verfahren wird dadurch nicht redundant: Die Notwendigkeit einer erweiterten Verschlüsselung des Abstimmungsverfahrens hängt von dem Sicherheitsbedarf eines Kunden ab. So kann davon ausgegangen werden, dass ein Freizeitverband weniger sensible Themen behandelt als ein Ausschuss der Hochschule. Das grundlegende Abstimmungsverfahren kann als Teil der Basisversion von OpenSlides erhalten werden. Kunden mit erhöhtem Sicherheitsbedarf werden durch die Crypto-Voting Erweiterung bedient. Öffentliche Verwaltungen haben im Bitkom Open-Source-Monitor 2023 den Fachkräftemangel (28%) und Sicherheitsaspekte (21%) als größte Nachteile von Freier Software angegeben (Schnaak und Termer 2023, S. 50). Beide Pain Points in einem konsolidierten Konzept zu lindern, bietet das größte Potenzial zur Akquise von Verwaltungen, die Freie Software bislang nicht einsetzen. Die Erweiterungen sollen regelmäßig wiederkehrende Zahlungen generieren, um langfristige Einnahmen zu erzielen. Hier sind Abonnements von Vorteil. In Bezug auf die Mindestlaufzeit sind sowohl jährliche als auch monatliche Modelle anzubieten. Zwar sind jährliche Abonnements zur Kundenbindung von Vorteil, jedoch ist dies als alleiniges Modell nicht empfehlenswert, da die Bereitschaft zur langfristigen Bindung bei neuen Modellen geringer ist. Bei einer zweigleisigen Laufzeit sind Kaufanreize für das jährliche Abonnement zu schaffen, beispielsweise in Form einer subadditiven Preishöhe gegenüber einem monatlichen Abonnement für ein Jahr, um Preisvorteile zu erhalten. Wie im Kapitel der Nutzwertanalyse vermittelt, birgt Freie Software inhärent die Möglichkeit, vom Nutzer heruntergeladen und selbst gehostet zu werden. Somit geht der ökonomische Nutzen für die Intevation GmbH verloren, da hieran kein Geld verdient wird. Unter der Betrachtung des Fachkräftemangels für Freie Software und dem Einarbeitungsaufwand für Öffentliche Verwaltungen entstehen aus den integrierten Serviceleistungen Kaufanreize sowie Lock-In-Effekte, da der Betrieb der Erweiterungen auf externen Kompetenzen basiert (Schnaak und Termer 2023, S. 50). In Kombination mit den Hosting-Paketen wird dieser Effekt verstärkt. Der Umfang der inbegriffenen Dienstleistungen ist auf die vorliegende Erweiterung abzugrenzen – im Falle von Selbsthoster von OpenSlides ist das Leistungsspektrum nur auf technischen Support im Ticket-System oder per Telefon zu begrenzen, um die komplette Bereitstellung als SaaS exklusiv zu halten. Somit können Upgrade-Anreize erschaffen werden und Cross-Selling zwischen Erweiterungen und Hosting-Paketen ermöglicht werden.

KONZEPT

Grundlegendes Konzept

Die Preisgestaltung von Software wird fortlaufend untersucht und im Zusammenspiel mit dem aktuellen Stand der Technik angepasst. Lehmann und Buxmann haben im Jahr 2009 Parameter von Preismodellen für Softwareprodukte definiert, basierend auf einer Fallstudie zur der SaaS-Preisgestaltung. Von besonderer Relevanz für diesen Artikel sind die Parameter zur Preisbildung und Preisermittlung.

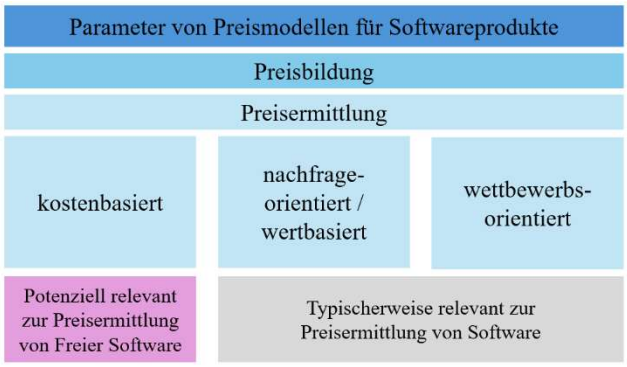


Abbildung 2: Einordnung von Freier Software in die Parameter zur Preisermittlung von Software
Eigene Abbildung in Anlehnung an: (Lehmann und Buxmann 2009, S. 520)

Die Preisbildung von Software orientiert sich in der Regel an der Nachfrage oder dem Wettbewerb für das Pro-

dukt, um insbesondere auf einem Markt mit vielen Substituten einen großen Marktanteil durch die Nutzung von Wettbewerbsvorteilen zu erreichen (Lehmann und Buxmann 2009, S. 521). Die Parameter tragen eine besondere Relevanz zur Erstellung des Konzeptes zur Preisermittlung von Freier Software, da die üblicherweise wettbewerbs- oder nachfrageorientierte Ermittlung für Softwarepreise im Kontext von Freier Software durch die Kostenbasis ergänzt wird.

Shapiro und Varian argumentieren gegen die kostenbasierte Preisermittlung, da sich der Preis von Informationsgütern (inkl. Softwarelizenzen) am Kundennutzen orientieren muss (Shapiro und Varian 1999, S. 3). Lehmann und Buxmann bemerkten, dass im Falle von SaaS die kostenorientierte Preisermittlung sinnvoll ist (Lehmann und Buxmann 2009, S. 521). Dies findet bei Servicekomplementen Relevanz, da diese als IT-Dienstleistungen aufgrund der klassischen Kostenstruktur nicht analog zu Software bepreist werden. Freie Software wurde in der Konzeption nicht berücksichtigt, da diese oftmals kostenfrei ist und primär durch ergänzende Leistungen Umsätze erzielt (Lehmann und Buxmann 2009, S. 519). Nach Shahrivar et al. werden die Entwicklungskosten vom Freier Softwarekern (FSK) durch die niedrigen Eintrittsbarrieren und somit hohen Verbreitung in der Entwicklung und Verwendung aufgefangen und aufgrund der typischerweise kostenfreien Bereitstellung eine de facto Kostenführerschaft erreicht (Shahrivar et al. 2018, S. 205–206). Diese Faktoren können sich positiv auf die Marktposition und den Verkauf von Komplementen auswirken. Jedoch sollte eine Möglichkeit bestehen, die entstandenen Entwicklungskosten der Freien Software im kommerziellen Kontext festzulegen, um sie durch MK auszugleichen. Die Richtlinien zu den Merkmalen von Freier Software sind dafür indikativ, dass die transitive Nutzung der Parameter zur Bepreisung von Freier Software Modifikationen benötigt, da klassische Monetarisierungsmodelle wie ein Lizenzverkauf nicht möglich sind. Jedoch findet die Kostenbasis in der Preisermittlung der MK Relevanz. Die Entwicklung von COSS obliegt im Vergleich zu OSS primär dem Unternehmen, welches die Freie Software in ihre Leistungserstellung integriert. Die Herstellung von Software ist durch hohe einmalige Fixkosten geprägt, da weitere Kopien mit keinen oder geringen Kosten verbunden sind (Frohmann 2022, S. 31). Durch die In-House-Entwicklung der Freien Software sind die Personalkosten während der Entwicklung dem Fixkostenblock zuzuordnen. Somit bietet sich für Freie Software die kostenbasierte Preisermittlung an, um a) die anfallenden Entwicklungskosten auf die Einzelkosten der Komplemente umzulegen und b) der Freien Software einen monetären Wert zuzuschreiben.

Umsetzung der Zuschlagskalkulation

Die untenstehende Abbildung veranschaulicht eine schematische Vorgehensweise zur Ermittlung von Zuschlagssätzen für Freie Software im Anwendungsfall vom COSS-Modell. In der Produktion werden die Herstellkosten anhand der Material- und Fertigungskosten ermittelt. Die Gemeinkosten werden prozentual auf die Kostenträger verteilt, um via Gewinnzuschlag den kalkulatorischen Nettoerlös für das jeweilige Produkt zu ermitteln (Hering 2014, S. 3–5). In einer summarischen Zuschlagskalkulation werden die gesamten Gemeinkosten mit den gesamten Einzelkosten dividiert, um die Gemeinkosten per gleichbleibenden Zuschlagssatz auf die Produkte zu verteilen (Kult o.J.). Eine prozentuale Gleichverteilung der Entwicklungskosten auf die MK wirft weitere Problematiken auf: Zum einen sollen MK mit höherer Komplexität, Kundenwert usw. einen entsprechend höheren Anteil der Entwicklungskosten tragen.

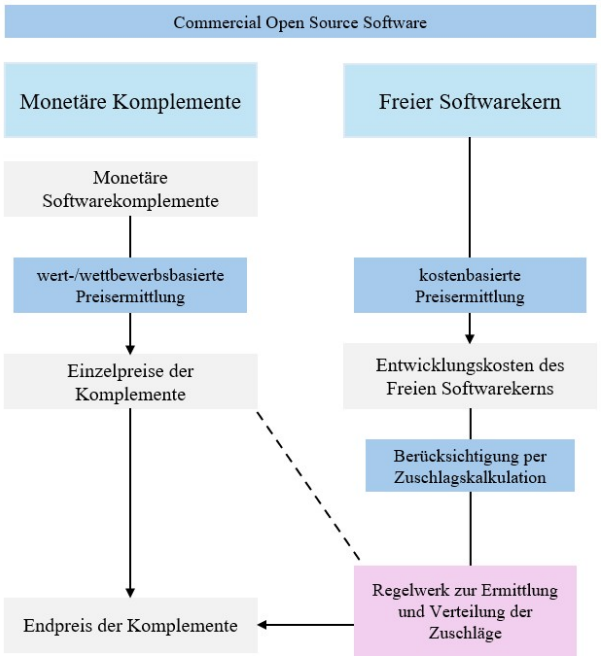


Abbildung 3: Schematische Vorgehensweise zur Zuschlagskalkulation im Fall von Commercial Open Source Software
Eigene Abbildung in Anlehnung an: (Lehmann und Buxmann 2009, S. 520; Shahrivar et al. 2018, S. 205; Hering 2014, S. 3)

Die Kalkulation der Zuschlagssätze wirft die Fragestellung auf, welche Basis der Freien Komplemente als Bemessungsgrundlage verwendet wird. Die folgenden Kapitel setzen sich mit der Konzipierung einer mehrdimensionalen Entscheidungsmatrix auseinander, anhand welcher fundierte sowie dynamische Zuschläge kalkuliert werden.

Methodik
Zur Ermittlung gewichteter Zuschlagssätze ist eine quantitative Vorgehensweise erforderlich, um die untersuchten Aspekte durch Analyseverfahren auf Basis der erhobenen Daten messbar und folglich vergleichbar zu

machen (Raithel 2008, S. 11). Um die Entscheidungsmatrix zur Bildung von gewichteten Zuschlagssätzen für die vorlegenden MK zu erstellen ist die Betrachtung aus der Innensicht und der Außensicht geplant. Die Innensicht befasst sich mit qualitativen Merkmalen aus dem Unternehmen, dem Projekt und den mit der Entwicklung verbundenen Prozessen. Die Außensicht berücksichtigt die Einstellung von Kunden zu den MK. Das Institute of Electrical and Electronics Engineers (IEEE) hat im IEEE Standard 1061 Softwarequalitätsmetriken definiert: "A function whose inputs are software data and whose output is a single numerical value that can be interpreted as the degree to which software possesses a given attribute that affects its quality" (IEEE Computer Society 1994, S. 3). Der Standard beinhaltet mehrere Metriken zur Bildung von Qualitätsfaktoren für die Bestimmung der Softwarequalität eines Systems (IEEE Computer Society 1994, S. 4). Auf Basis der Metriken dieses Standards wurden qualitative Merkmale von Software (QM) zur Bewertung der MK aus der Innen- und Außensicht ausgewählt:

1. Komplexität
2. Zeitaufwand
3. Ressourcenaufwand
4. Anzahl an Iterationen
5. Kundenzufriedenheit
6. Zahlungsbereitschaft

Innensicht

Line of Code (LOC) ist eine konventionelle Softwaremetrik und statische Produktmetrik zur Messung der Komplexität (Augsten 2019). Durch die Messung der Zeilenanzahl im Quellcode einer Software können Aussagen über die Komplexität der Software und dem Aufwand des Projekts getroffen werden (Spasojevic 2024). Eine für die Programmiersprache hohe Zeilenanzahl kann korrelativ zu den eingesetzten Zeit und Ressourcen für das MK sein und deutet auf eine hohe Komplexität hin (Spasojevic 2024). Die Prozessmetrik befasst sich mit dem Entwicklungsprozess anhand der Kennzahlen Anzahl an Fehlern, Anzahl an Änderungen, Ressourcen- und Zeitaufwand (Augsten 2019). Die separate Betrachtung ist zur mehrschichtigen Beurteilung der Komplexität relevant, da eine syntaxbedingt hohe Zeilenanzahl nicht unmittelbar eine hohe Komplexität bedeutet. Gleichwohl kann ein Projekt mit einer niedrigen Zeilenanzahl aufgrund von intensiver Recherche vor dem Schreiben des Quellcodes oder häufigen Korrekturen zeitintensiv sein. Äquivalent zur Ermittlung der Entwicklungskosten des FSK ist der Zeitaufwand durch die Dauer für das MK-Projekt und der Ressourcenaufwand durch die Anzahl an Entwicklern messbar. Die Anzahl an Iterationen konsolidiert notwendige Änderungen am Programm aufgrund von Fehlern oder anderen Modifikationen. Zur Analyse des Quelltextes und der Feststellung von Defekten bestehen zwei Methoden: Manual Code Review und Static Code Analysis (Stefanovic et al. 2020, S. 566). Die manuelle Überprüfung erfordert Personal, Arbeitsstunden und projektgebundenes Know-how, während eine statische Analyse des Quelltextes auf Basis von eigenen oder externen Tools erfolgt (Stefanovic et al. 2020, S. 566). Eine Static Code Analysis ist im Hinblick auf die Skalierbarkeit von Projekten günstiger, da MK mit wachsender Komplexität exponentiell mehr Zeit zur manuellen Überprüfung in Anspruch nehmen. Der Zyklus ist in vier Hauptphasen eingeteilt: 1. Establish goals 2. Run the static analysis tool 3. Review code (using output from the tool) 4. Make fixes (Chess und West 2007, S. 47–70). Chess und West prognostizieren auf Basis von mehreren Problemen in der ersten Iteration auf Fehler in Folgeiterationen (Chess und West 2007, S. 47–70). Anhand der benötigten Durchläufe des Code Review Cycle bis zur Veröffentlichung des vorliegenden MK wird die benötigte Menge an Fehlerbehebungen quantifiziert.

Außensicht

Die Außensicht bezieht sich auf die Wahrnehmung der Kunden, um eine vollständige Bewertung der MK zu erzielen. Zur Erhebung ist die Erstellung eines geschlossenen Fragebogens sinnvoll, da standardisierte Fragen a) die Quantifizierung der Ergebnisse für die Entscheidungsmatrix und b) die Vergleichbarkeit der Ergebnisse für die MK gewährleistet (Reinders 2011, S. 53–59).
Die Kundenzufriedenheit befasst sich mit der Bewertung von Produkten oder Dienstleistungen in Bezug auf die (Nicht-) Erfüllung von Erwartungen (Kirchgeorg 2018). Die Kennzahl wird im Fragebogen mit dem Customer Satisfaction Score (CSAT) erhoben, in welchem die Zufriedenheit mit dem Produkt oder Teilaspekten als Likert-Item auf einer Skala von 1 (sehr unzufrieden) bis 5 (sehr zufrieden) erfragt wird (Föhl und Friedrich 2022, S. 47–48). Aus den gesamten Ergebnissen wird der Mittelwert gebildet. Der subjektive Nutzen des MK aus Kundensicht ist aus der Differenz zwischen der Zahlungsbereitschaft und dem Preis zu ermitteln (Schäfers 2004, S. 9–10). Schäfers hat folgende Formel festgestellt (Schäfers 2004, S. 10):

$$KR_i = ZB_i - p = \begin{cases} \geq 0, \text{ falls i-ter Nachfrager das Produkt kauft} \\ = 0, \text{ falls i-ter Nachfrager das Produkt nicht kauft} \end{cases} \quad (i \in I), \tag{1-0}$$

wobei:
KR_i: Konsumentenrente des i-ten Nachfragers,
p: Preis des Produktes,
ZB_i: Zahlungsbereitschaft des i-ten Nachfragers,
I: Indexmenge der Nachfrager.

Unternehmen können eine hohe positive Konsumentenrente nutzen, um höhere Zahlungsbereitschaft zu fördern, während eine niedrige oder negative Konsumentenrente diese mindert, was oft zum Verzicht auf den Kauf führt (Schäfers 2004, S. 10). Die Zahlungsbereitschaft sollte direkt im Fragebogen ermittelt werden. Der Durchschnitt der Antworten wird vom Produktpreis subtrahiert, um die durchschnittliche KR zu erfahren. Auf Basis von (1-0) werden folgende Formeln abgeleitet:

$$KR_\emptyset = ZB_\emptyset - p \tag{1-1}$$

$$= \begin{cases} \geq 0, \text{ falls } \emptyset\text{Nachfrager das Produkt kauft} \\ = 0, \text{ falls } \emptyset\text{Nachfrager das Produkt nicht kauft} \end{cases} \quad (i \in I),$$

Das Verhältnis zum Preis ergibt einen Score zur Gewichtung.

$$QM_{ZB} = \frac{p}{\emptyset KR} \tag{1-2}$$

Fragebogen

Zur Erstellung des Fragebogens sind die Eigenschaften Zielgruppe, Form und Dauer zu beachten.

Als Zielgruppe sind allgemein Kunden von MK anvisiert. Näher wird der Fragebogen auf ein MK eingegrenzt, also ist der Fragebogen an Kunden des betrachteten MK adressiert. Da die Nutzung mehrerer MK pro Kunde möglich ist, besteht das Risiko einer „Überflutung" mit Fragebögen, was sich negativ auf die Rücklaufquote auswirken kann. Daher ist es ratsam, den Fragebogen mittels „Chunking" in identische Module aufzuteilen (Eberl 2016, S. 222). Eine Modularisierung auf Befragtenebene ermöglicht dem Befragten, relevante Module (hier: MK) des Fragebogen auszuwählen und durch Pausen- oder Beendigungsoptionen den Fragebogen nach einem Modul zu beenden (Eberl 2016, S. 222). Die Fragen pro Modul sind kurz zu halten. Kurze digitale Befragungen weisen mehrere Vorteile auf: Eine geringere Abbruchquote und höhere Bereitschaft auf Seite der Befragten sowie eine kürzere Dauer von der Konzeption bis zur Realisierung (Eberl 2016, S. 220). Zudem hat die Modularisierung auf Befragtenseite den Vorteil, dass die Befragungsdauer durch den Befragten bestimmt wird. (Eberl 2016, S. 222–223). Die folgende Abbildung stellt ausschnittsweise einen Fragebogen zur Messung der genannten QM dar.

Fragebogen zur Messung der Kundenzufriedenheit

Zur Sicherung Ihrer Zufriedenheit möchten wir Ihre Meinung zu den folgenden Produkten hören.

Abschnitt 0: Produktnutzung

Welche Produkte benutzen Sie aktuell oder haben Sie in der Vergangenheit benutzt?

(x) Produkt A (x) Produkt B () Produkt C (x) Produkt D () Produkt E () Produkt F

Modul für [Produkt A]

Abschnitt 1: Zufriedenheit mit dem Produkt

Wie zufrieden sind Sie insgesamt mit [Produkt A]?

() 5 Sehr zufrieden () 4 Zufrieden () 3 Neutral () 2 Unzufrieden () 1 Sehr unzufrieden

Abschnitt 2: Zahlungsbereitschaft

Welchen Preis halten Sie für [Produkt A] auf Basis von Ihrer Nutzung angemessen?

_____€

☐ Weiter ☐ Pausieren ☐ Fertigstellen

Modul für [Produkt B]

(...)

Abbildung 4: Ausschnitte eines Fragebogens zu Messung der Kundenzufriedenheit
Eigene Abbildung

Entscheidungsmatrix

Die Entscheidungsmatrix ist eine Methode zur Konsolidierung der festgestellten QM für die MK in eine Tabelle, um mittels eines Scorings auf Basis von messbaren Daten begründete Entscheidungen zur Vergabe der Zuschlagssätze für die Entwicklungskosten des FSK zu treffen. Sie visualisiert die Gewichtungen für alle MK auf verständliche Art und Weise und ist ein Instrument zur Verwendung der vorangehenden individuellen Berechnungen und Erhebungen.

Die jeweiligen QM werden gewichtet, und für die jeweiligen Produkte werden Scores auf einer Skala von 1 (sehr niedrig) bis 10 (sehr hoch) vergeben. Grundsätzlich ist die Verteilung der Gewichtungen von der Wahrnehmung und Strategie des Unternehmens abhängig. Für manche sind prozessmetrikbezogene QM der Innensicht von höherer Bedeutung, während andere den Fokus auf die Außensicht legen. Die Gewichtung ist also in Absprache mit dem Unternehmen zu treffen. Auf Basis der vorausgehenden Erkenntnisse über die QM wurde die Gewichtung der Innen- und Außensicht jeweils mit 50% festgelegt. Es wurde festgestellt, dass die Zahlungsbereitschaft entscheidend für die Kaufentscheidung ist, da eine negative Konsumentenrente einen Kauf verhindert. Daher ist diesem Kriterium mit 30% eine hohe Relevanz zuzuschreiben, da es die Obergrenze für die Höhe des Zuschlagssatzes bestimmt. Die Kundenzufriedenheit wird entsprechend mit 20% bewertet. Die Komplexität liefert Erkenntnisse über den Umfang der Software und die Intensität der Funktionsweise und ist daher mit 20% zu bewerten. Zeitaufwand, Ressourcenaufwand und die Anzahl der Iterationen beziehen sich auf die Prozessmetrik und werden jeweils mit 10% bewertet, da sie dasselbe Merkmal aus unterschiedlichen Perspektiven messen. Damit die gewichteten Summen zur Bestimmung der gewichteten Zuschlagssätze verwendet werden können, müssen die Summen anschließend normalisiert werden, um das Gesamte der Entwicklungskosten des FSK abzubilden. Die Vergabe der Scores erfolgt zunächst in Relation zum gegenwärtigen Wissensstand und dem spezifischen Produktspektrum. Die Matrix erfasst somit statisch einen bestimmten Zeitpunkt. Zukünftig sollten die Matrizen dynamisch gestaltet werden, sodass sie auf Grundlage neuer Erkenntnisse iterativ und in regelmäßigen Intervallen angepasst werden.

Tabelle 5: Entscheidungsmatrix zur Bestimmung von Zuschlagssätzen
Eigene Abbildung

		QM	Gew.	MK A		MK B		MK C	
				S	**gS**	S	**gS**	S	**gS**
Innensicht	LOC CRC Personal Projektdauer	Komplexität	0,2	10	**2,0**	3	**0,6**	10	**2,0**
		Zeitaufwand	0,1	8	**0,8**	2	**0,2**	2	**0,2**
		Ressourcenaufwand	0,1	2	**0,2**	1	**0,1**	(…)	
		Anzahl Iterationen	0,1	9	**0,9**	3	**0,3**		
Außensicht	CSAT Konsumentenrente	Kundenzufriedenheit	0,2	7	**1,4**	6	**1,2**		
		Zahlungsbereitschaft	0,3	5	**1,5**	1	**0,3**		
		gew. Summe	1	/	**6,8**	/	**2,7**		

Dynamisches Konzept zur Preisfeststellung

Dynamic Pricing wird in Branchen wie der Hospitality, dem Textileinzelhandel oder dem Flugverkehr zur dyna-

mischen Preisgestaltung von Produkten oder Dienstleistungen auf Basis von verschiedenen Kriterien verwendet. Ungeachtet der starken Heterogenität der Definitionsansätze werden vereinfachend unter Dynamic Pricing Preisvariationen auf Basis vom Zeitverlauf oder personenbezogener Merkmale verstanden (Priester 2022, S. 13). In der Produktionsbranche kann festgestellt werden, dass Dynamic Pricing aufgrund der Erfolge in der Erzielung einer höheren Rendite eine jährliche Zuschlagskalkulation ersetzt (Oslak 2023). Die Problematik in der Übertragung auf Software und Peripherie liegt in den fundamentalen Unterschieden der Preisgestaltung. Ein prägnantes Beispiel für Dynamic Pricing ist die Preisabhängigkeit von Textilwaren auf Basis der Saison sowie dem Wetter (Priester 2022, S. 19). Eine Winterjacke wird im Juni günstiger erhältlich sein als im November. Der Preisanpassungszyklus von Software weist größere Intervalle auf: Beispielsweise führt IBM jährliche Preisanpassungen im Rahmen der General Price Harmonization durch, jeweils geltend zum 1. Januar des kommenden Jahres (CURSOR Software AG 2022, 2023, 2024). Die Intervalle werden von Unternehmen zu Unternehmen variieren, jedoch kann aufgrund der unterschiedlichen Struktur von typischen dynamisch bepreisten Produkten und Software das Konzept nicht 1:1 angewandt werden. Trotzdem bietet sich die regelmäßige Überprüfung der QM aus der Entscheidungsmatrix für die gewichteten Zuschlagssätze aus mehreren Gründen an. Zunächst ist das Wachstum des Produktportfolios zu berücksichtigen: Neue MK werden bei Aufnahme in die Entscheidungsmatrix die Gewichtungen verändern, was sich auf alle Preise auswirkt. Zudem ist die Orientierung an der Nachfrage zur Preisbildung im Rahmen einer iterativen Konzeption zu berücksichtigen. Die Nachfrage für ein MK kann aufgrund von externen Faktoren positiv wie negativ beeinflusst werden. Dies wirkt sich quantitativ messbar auf die QM der Entscheidungsmatrix aus. Beispielsweise kann die Zahlungsbereitschaft durch ökonomische Faktoren wie eine Rezession negativ beeinflusst werden, was Handlungsbedarf seitens des Unternehmens indiziert.

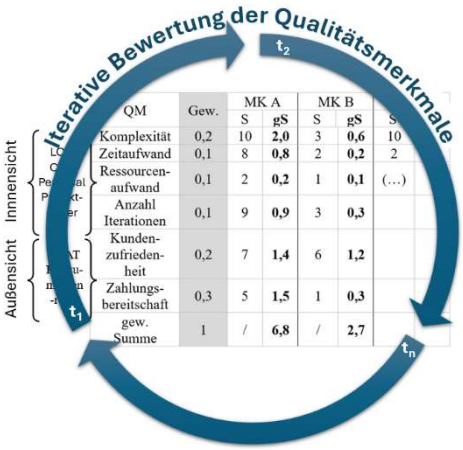


Abbildung 5: Konzept zur dynamischen Preisbildung der monetären Komplemente
Eigene Abbildung

Die Intervalllänge ist unter Berücksichtigung der individuell bestehenden Preisanpassungsprozesse des Unternehmens zu definieren.

Validierung des Konzeptes mit OpenSlides

Das Konzept zur Preisermittlung von MK unter Berücksichtigung der Entwicklungskosten vom FSK via gewichteten Zuschlagssätzen ist zum aktuellen Stand ein theoretisches Konstrukt, welches unter Erfolgsaussichten in die Praxis (das operative Geschäft der Intevation GmbH) umgesetzt werden soll. Um eine Validierung vorzunehmen, wird anhand von Sekundärdaten eine Testumgebung erstellt, in welcher übertragbare Erkenntnisse erzielt werden. Dies besitzt für die Strategie der Intevation GmbH große Relevanz, da eine Entscheidung für ein neues Konzept mit Opportunitätskosten, primär in Form von Zeit und Personal, verbunden ist. Dazu sind folgende Schritte notwendig:

1. Berechnung der Kennzahlen für die QM
2. Bestimmung der Gewichtungen
3. Vergabe der Scores auf Basis der Kennzahlen
4. Ermittlung der gewichteten Summen
5. Normalisierung der gewichteten Summen
6. Ermittlung der Zuschläge aus dem Kostenblock Entwicklungskosten des FSK für die MK
7. Bildung eines Zeithorizonts t_n bis zum Ausgleich des Zuschlags
8. Prognose der Absatzzahlen zur Verteilung der Entwicklungskosten des FSK für t_n

Kundensegmente

Die aktuellen Kunden von OpenSlides sind mehrheitlich öffentliche Einrichtungen, die aus heterogenen Teilnehmern bestehen und durch regelmäßige Treffen den Zweck ihrer Organisation verfolgen. Dazu gehören Gewerkschaften, Kirchen, Parteien, Verbände, Vereine, Hochschulen und studentische Organisationen (OpenSlides-Team o. J.b). Als neue Zielgruppen sind Landtage und Betriebsräte anvisiert, welche zu den öffentlichen Einrichtungen gehören (Intevation GmbH 2024). Öffentliche Verwaltungen zeigten sich im Bitkom Open Source Monitor 2023 zum Thema Freie Software (in der Studie: Open-Source-Software) zu 14% sehr aufgeschlossen, 26% eher aufgeschlossen, 15% eher ablehnend und 4% sehr ablehnend – 35% waren unentschieden (Schnaak und Termer 2023, S. 48). Öffentliche Verwaltungen wurden gezielt in den Veröffentlichungen aus den Jahren 2023 und 2021 befragt. Zum Vergleich der Ergebnisse aus dem Jahr 2023 mit 2021 wurden die Kategorien „sehr aufgeschlossen“ mit „eher aufgeschlossen“ sowie „eher ablehnend“ mit „sehr ablehnend“ konsolidiert.

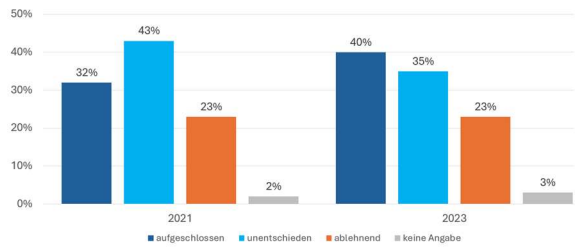


Abbildung 6: Haltung Öffentlicher Verwaltungen zum Thema Freier Software
Eigene Abbildung in Anlehnung an (Gentemann et al. 2021, S. 51; Schnaak und Termer 2023, S. 48)

Der Vergleich verdeutlicht zwei Effekte: Die Haltung der Öffentlichen Verwaltung zu Freier Software entwickelt sich positiv, was pauschal ein vergrößertes Kundensegment bedeutet. Dies wird unterstützt durch die günstigen Bedingungen für Freie Software in der Öffentlichen Verwaltung: Im Rahmen des Digitalpolitischen Dossier im Deutschen Bundestag wurde im Jahr 2019 zur Unterstützung und Neubewertung von Freier Software in der Verwaltung aufgerufen, um eine „digitale Souveränität" zu erreichen. Es wurden die Abhängigkeit von einzelnen Anbietern und Datenschutzbedenken gegenüber Cloud-Lösungen kritisiert, unter anderem da der Zugriff des Anbieters durch die Verschleierung von proprietären Quellcode intransparent ist (Kompetenzzentrum Öffentliche IT 2019). Digitale Souveränität für öffentliche IT-Nutzer bedeutet im Wesentlichen, sensible Daten zu schützen und IT-Systeme bedarfsgerecht und effizient zur Verfolgung der Ziele zu nutzen (Goldacker 2017, S. 8). Das BSI positioniert sich in dieser Angelegenheit zur Freien Software und stellt die Vorteile dar, insbesondere in Bezug auf Anpassbarkeit und Sicherheit – unter der Bedingung, dass Kompetenzen zur IT und Freien Software vorhanden sind (Bundesinstitut für Sicherheit in der Informationstechnik o. J.). Auf Bundesebene wurden im Jahr 2022 finanzielle Maßnahmen zur Stärkung der digitalen Souveränität mit Freier Software erlassen (Krempl 2022). Solche Effekte beeinflussen nicht nur die grundlegende Nachfrage, sondern auch die Zahlungsbereitschaft, was sich unmittelbar auf die Ermittlung der QM-Scores zur Bildung gewichteter Zuschlagssätze auswirkt. Die wahrgenommenen Vor- und Nachteile für den Einsatz von Freier Software in der Öffentlichen Verwaltung sind weitere Indikatoren zur Ermittlung der Zahlungsbereitschaft. Im Open Source Monitor 2019 und 2021 wurden Öffentliche Verwaltungen dazu nicht gesondert befragt. Zur Gewährleistung der Vergleichbarkeit wird die Haltung von Unternehmen zu Freier Software in den Jahren 2019, 2021 und 2023 verglichen und die Haltung der Öffentlichen Verwaltung im Jahr 2023 separat betrachtet, um Veränderungen auf Basis der vorausgegangenen Erkenntnisse abzuleiten.

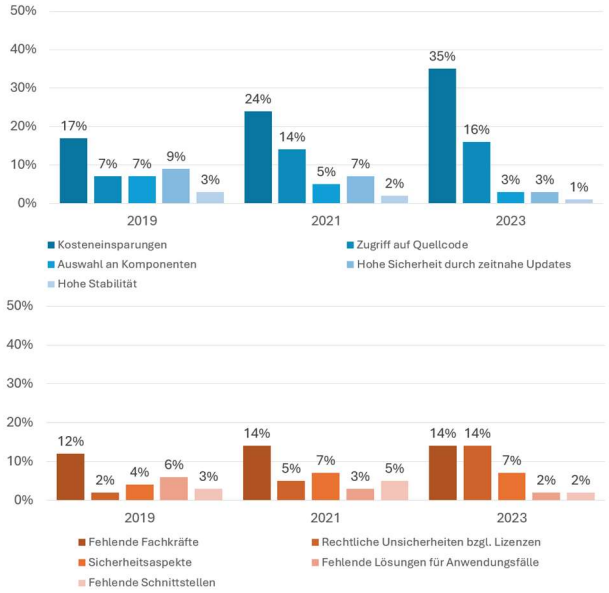


Abbildung 7: Vorteile und Nachteile für den Einsatz von Freier Software im Unternehmen 2019 bis 2023
Eigene Abbildung in Anlehnung an: (Gentemann und Termer 2020, S. 22–23; Gentemann et al. 2021, S. 13–14; Schnaak und Termer 2023, S. 49–50)

Der Vergleich bestätigt positive wie negative Schwankungen der befragten Kategorien im Zeitverlauf. Kosteneinsparungen wurden im betrachteten Zeitraum erhöhte Relevanz zugeschrieben. Auf Basis dieser Datenlage sind zwei Szenarien zu erstellen: (1) Aufgrund von Kosteneinsparungen durch den Einsatz von Freier Software sind IT-Budgets freigeworden, welche für den Erwerb von MK verwendet werden können. Die in den Fragebögen angegebene Zahlungsbereitschaft steigt. (2) Kosteneinsparungen sind durch verknappte IT-Budgets aufgrund von erhöhten Ausgaben in anderen Geschäftsbereichen notwendig. Die Zahlungsbereitschaft sinkt. Je nach Szenario sind die Scores des QM in der Entscheidungsmatrix (1) zu erhöhen oder (2) zu verringern. Das Monitoring dieser Haltungen ist nicht nur zur iterativen Bewertung der QM relevant, sondern auch zur nachfrageorientierten Preisermittlung von MK. Öffentliche Verwaltungen sehen die größten Vorteile in Kosteneinsparungen (18%), Zugriff auf den Quellcode und Auswahl an Komponenten (jeweils 11%), während fehlende Fachkräfte (28%), Sicherheitsaspekte (21%) und rechtliche Unsicherheiten (9%) als größte Nachteile wahrgenommen werden (Schnaak und Termer 2023, S. 49–50).

Die Entscheidungsmatrix wird auf Basis des MK zur Blockchain-Verschlüsselung von Abstimmverfahren (hier genannt: MK „Crypto") aufgebaut. In der Literatur wurde ein Blockchain-Technologie-Abstimmungsverfahren als Vorhaben mit hoher Komplexität festgestellt, welches Audits und weitere Prüfverfahren zur Sicherstellung der Effektivität und Verlässlichkeit benötigt (Stanciu et al. 2023, S. 6). Aus der Innensicht sind die QM Komplexität mit dem Score 8, der Zeit- und Ressourcenaufwand auf Basis der betroffenen Kennzahlen Personal und Projektdauer jeweils mit dem Score 7 zu bewerten. Die Anzahl an Iterationen wird aufgrund von be-

nötigten Prüfverfahren mit anschließenden Anpassungen, welche sich im Code Review Cycle manifestieren, mit dem Score 10 bewertet. Auf Basis der wettbewerbs- und nachfragebasierten Preisermittlung wird in Orientierung an WooCommerce Marketplace Extensions der Höchstpreis von $299 festgesetzt (Woo o. J.h) (Letzte Überprüfung im November 2024). Bei öffentlichen Verwaltungen mit erhöhtem Sicherheitsbedarf zur Durchführung von Versammlungen wird von einer erhöhten Zahlungsbereitschaft für dieses MK ausgegangen. Zur Veranschaulichung der Struktur der prognostizierten Zahlungsbereitschaft wurde eine abstrahierte Hypothese aufgestellt, mit der folgenden Annahmelogik: Es wird einer 67,2%-igen Steigerung des zuvor festgelegten MK-Preises unterstelltauf $500 festgesetzt. Die Steigerung ergibt sich jeweils mit circa 1/3 der Steigerung für die Befriedigung des hohen Sicherheitsbedarfes, dem Innovationsgrad des eingesetzten Blockchain-Verfahrens sowie dem zur Durchführung notwendigen Service. Zudem wurde sich für eine Zahlungsbereitschaft von $500 statt $478 (im Falle einer 60%-igen Steigerung) entschieden. Aufgrund der angewandten Methode zur Ermittlung der Zahlungsbereitschaft im Fragebogen ist aufgrund der explorativen Natur der Open-Ended-Methode nicht zu erwarten, dass von den Ansprechpersonen exakte Preisvorstellungen abgegeben werden, sondern eher gerundete Größenordnungen (Müller et al. 2022, S. 10–15). Zur Ermittlung des Scores für das QM „Zahlungsbereitschaft“ wird anhand von Formel (1-1) die durchschnittliche Konsumentenrente ermittelt,

$$KR_{\emptyset} = \$201 = \$500 - \$299 \tag{1-1}$$

welche anschließend in Relation zum Preis gesetzt wird:

$$QM_{ZB} = 0{,}672 = \frac{\$201}{\$299} \tag{1-2}$$

Da die Konsumentenrente 67,2% des Originalpreises darstellt, wird das QM „Zahlungsbereitschaft“ mit dem Score 7 bewertet. Da das MK den zu 21% genannten Nachteil „Sicherheitsaspekte“ bedient, wird eine hohe Kundenzufriedenheit von 8 unterstellt. Um die Verteilung der gewichteten Zuschlagssätze zu veranschaulichen, wird das MK „Test“ mit zufälligen Scores erstellt. Zur Normalisierung werden die Scores summiert und jeweils durch die Gesamtsumme geteilt.

Tabelle 6: Beispielhafte Entscheidungsmatrix
Eigene Abbildung

QM	Gew.	MK „Crypto”		MK „Test”	
		S	**gS**	S	**gS**
Komplexität	0,2	8	**1,6**	4	**0,8**
Zeitaufwand	0,1	7	**0,7**	3	**0,3**
Ressourcenaufwand	0,1	7	**0,7**	2	**0,2**
Anzahl Iterationen	0,1	10	**1**	5	**0,5**
Kundenzufriedenheit	0,2	8	**1,6**	6	**1,2**
Zahlungsbereitschaft	0,3	7	**2,1**	2	**0,6**
gew. Summe	1	/	**7,7**	/	**3,6**
Normalisiert	/	$\frac{7,7}{11,3} = 0,681$		$\frac{3,6}{11,3} = 0,319$	

Im Zeitverlauf verschieben sich die gewichteten Summen aufgrund von Veränderungen der Datenlage, was die gewichteten Zuschlagssätze aus der Normalisierung beeinflusst.

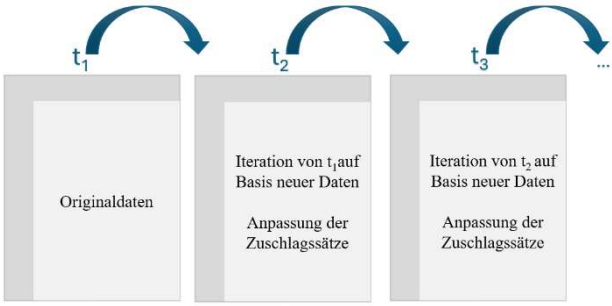


Abbildung 8: Iterationsschema der Entscheidungsmatrizen
Eigene Abbildung

Das Abonnement-Modell der MK für OpenSlides Plus verkompliziert die Zuteilung der Entwicklungskosten des FSK per Zuschlagssatz. Sollten die gesamten Entwicklungskosten per FSK auf eine monatliche Rate zugeschlagen werden, wären diese in der Theorie nach einem Monat ausgeglichen. Jedoch ergeben sich hieraus drastisch höhere Preise für die MK, welche die Kunden im Hinblick auf das QM Zahlungsbereitschaft bei negativer Konsumentenrente nicht bereit zu zahlen sind. Die Konsumentenrente ist als Obergrenze zur Verteilung der Entwicklungskosten des FSK zu sehen. Liegen die Entwicklungskosten beispielsweise bei 10.000€ und die zwei MK aus der Entscheidungsmatrix sind als Kostenträger vorhanden, trägt das MK „Crypto“ in t_1 6.810€ und das MK „Test“ 3.190€. Wenn t_1 ein Jahr ist, dann ist der Kostenblock auf 12 Monate zu verteilen (6.810€/12 Monate = 567,50€). Das MK „Crypto“ hat pro Monat 567,50€ auszugleichen. Jedoch indiziert die Konsumentenrente aus dem QM Zahlungsbereitschaft, dass Konsumenten zum Befragungszeitpunkt im Durchschnitt zur zusätzlichen Zahlung von $201 bzw. 185€ bereit sind. Die Ausschöpfung der Konsumentenrente bedeutet jedoch, dass ein geringerer Spielraum für weitere Preisanpassungen bestehen, da die Konsumentenrente sich nicht zwingend linear positiv entwickelt. Daher ist a) der Zeithorizont (ein Ausgleich über 36 Monate würde 189,76€/Monat bedeuten) und b) das Verkaufsvolumen (10 Jahresabonnements reduzieren den 12-Monatsblock auf 56,75€/MK) zu berücksichtigen. Prognosen zum Absatz können anhand von Sekundär- oder Primärdaten getroffen werden. Primärdaten bieten direkte Zahlen: Vorregistrierungen für das MK via Newsletter oder Fragebogen messen das grundlegende Interesse an einem MK, was Schätzungen zum Absatz ermöglicht. Hier sind Kombinationen der Vorgehensweisen zur Bildung einer mehrdimensionalen Datenlage möglich. Ausgehend von den Sekundärdaten bietet der Bitkom Open-Source-Monitor Anhaltspunkte.

Unter der Annahme, dass die Teilsegmente Gewerkschaften, Gesundheitswesen und Hochschulbereich Bedarf am MK „Crypto“ haben, wird auf ein Maximum von 35 Kunden geschlossen (OpenSlides-Team o. J.b). 21% der befragten Öffentlichen Verwaltungen haben im Open-Source-Monitor Sicherheitsaspekte von Freier Software bemängelt (Schnaak und Termer 2023, S. 50). Dies bedeutet 7,35 bzw. 7 der 35 Kunden als Minimum. Der Mittelwert ist 21 Kunden, was einen Zuschlag der Entwicklungskosten des FSK für t_1 von 27,02€ ergibt (567,50€/21 Kunden). Dieselbe Rechnung ist für weitere MK durchzuführen.

Interpretation der Ergebnisse

Die Validierung zeigt auf, dass die theoretischen Konstrukte aus der Konzipierung Werte ausgeben, welche die Ermittlung des Preises von MK unter Berücksichtigung des FSK beeinflusst. Theoretische Ergebnisse können durchaus von Parametern der Praxis abweichen – Erfahrungen und betriebswirtschaftliche Kenntnisse sind als ergänzende Ebene heranzuziehen, um die Auswirkungen der Zuschläge auf den Originalpreis des MK auf die Eignung zu überprüfen. Dazu sind die zuständigen und/oder leitenden Positionen aus dem Unternehmen heranzuziehen, um ergänzendes Know-how sowie Sichtweisen zu erlangen. Noch vor der Kalkulation ist der Zeithorizont bis zum Erreichen des Break-Even-Points für die Entwicklungskosten des FSK zu ermitteln, nicht nur zum Beeinflussen der Verteilung, sondern auch als Horizont bis zum Erreichen der Kennzahl als Bestandteil des strategischen Controllings. Nach der Kalkulation ist der Anteil der Entwicklungskosten des FSK am Endpreis des MK zu bewerten. Trotz der gewichteten Scores kann durch hohe Entwicklungskosten oder hohe Scores in einzelnen Kategorien der Zuschlag den wert-/nachfrageorientiert ermittelten Preis überschreiten. Sofern ein wesentlicher Anteil (z.B. >50% des Endpreises) durch die Entwicklungskosten verursacht wird, ist die Verhältnismäßigkeit des Zuschlags fraglich. Selbst wenn der Endpreis innerhalb der Konsumentenrente liegt, ist über die anteilige oder komplette Umlage der Entwicklungskosten des FSK für dieses MK zu entscheiden. MK mit hoher Konsumentenrente können den Vorteil aufweisen, dass sie als „gutes Angebot“ als Eintrittskarte für das neue Monetarisierungskonzept gelten können. Zudem ist aus organisatorisch-strategischer Sicht ein Zeitplan zur Iteration der Entscheidungsmatrix aufzustellen sowie fortlaufende Überprüfungen, ob die Implementierung erfolgreich verläuft.

FAZIT UND AUSBLICK

Die Implementierung von Freier Software in die betriebliche Leistungserstellung weist verschiedene Faktoren auf, die im Vergleich zu proprietärer Software zu berücksichtigen sind. Dies manifestiert sich in zwei Hauptaspekten: Der Konzipierung des Monetarisierungsmodells sowie der Preisbildung. Das Monetarisierungsmodell befindet sich in einem Spannungsfeld zwischen der Berücksichtigung der charakteristischen Freiheiten von Freier Software und der Erzielung von Umsatz. Das COSS-Modell zeigt Wege zur Koexistenz auf und kann als Grundlage zur Ermittlung neuer Wege verwendet werden. Die Preisbildung befasst sich mit der Problematik, dass die Entwicklungskosten des FSK keine Berücksichtigung in der Monetarisierung der MK finden. Zudem findet die kostenbasierte Preisermittlung für den FSK neue Relevanz. Durch die Verteilung der Entwicklungskosten via gewichteten Zuschlagssätzen auf die MK werden diese in der Preisbildung berücksichtigt. Ein iteratives Konzept zur konstanten Neubewertung der Zuschlagssätze ermöglicht dieselbe Dynamik in der Preisgestaltung wie die Orientierung am Wettbewerb oder dem Wert. Über diesen Artikel hinaus ist zu untersuchen, welche Möglichkeiten künstliche Intelligenz auf die Automatisierung der Bildung sowie Iteration der Entscheidungsmatrix bergen. Hier ist zu bemerken, dass die menschliche Komponente in der Interpretation der Ergebnisse unabdingbar ist. Das Konzept wurde am Beispiel von OpenSlides validiert, um die Vorgehensweise zur Implementierung zu verdeutlichen. Die Konditionen stellten eine Testumgebung dar, um die Entscheidung für oder gegen die Umsetzung des Konzeptes zu erleichtern. Eine Weiterführung der Forschung in der Praxis wird die tatsächliche Validierung des Konzeptes ermöglichen. Dazu kann das Monetarisierungskonzept als Beta-Version in kleinem Rahmen durchgeführt werden.

ABKÜRZUNGSVERZEICHNIS

COSS	Commercial Open Source Software
FSK	Freier Softwarekern
MK	Monetäre Komplemente
OSS	Open Source Software
QM	Qualitative Merkmale

LITERATUR

Augsten, Stephan (2019): Was sind Softwaremetriken? In: *Dev-Insider*, 26.04.2019. Online verfügbar unter https://www.dev-insider.de/was-sind-softwaremetriken-a-813487/, zuletzt geprüft am 15.02.2025.

Brassel, Stefan; Gadatsch, Andreas (2019): Softwarelizenzmanagement kompakt. Einsatz und Management des immateriellen Wirtschaftsgutes Software und hybrider Leistungsbündel (Public Cloud Services). 1. Auflage 2019. Wiesbaden: Springer Fachmedien Wiesbaden GmbH; Springer Vieweg (IT kompakt).

Bundesinstitut für Sicherheit in der Informationstechnik (o. J.): FLOSS (Free/Libre Open Source Software). Strategische Position des BSI. Online verfügbar unter https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-

Empfehlungen/Freie-Software/freie-software_node.html, zuletzt geprüft am 04.11.2024.

Cheng, Shenghui (2024): Web 3.0: Concept, Content and Context. 1. Aufl.: Springer.

Chess, Brian; West, Jacob (2007): Secure programming with static analysis. Upper Saddle River, NJ, Munich: Addison-Wesley (Software security series).

Comino, Stefano; Manenti, Fabio M. (2011): Dual licensing in open source software markets. In: *Information Economics and Policy* 23 (3-4), S. 234–242. DOI: 10.1016/j.infoecopol.2011.07.001.

CURSOR Software AG (2022): IBM Preisanpassungen für Software 01.01.2023. Update 22. November 2022. Online verfügbar unter https://www.cursor-distribution.de/de/home-de/community-informix/informix-news-details/1378-ibm-preisanpassungen-software-1-jan-2023-update, zuletzt geprüft am 02.03.2025.

CURSOR Software AG (2023): IBM Preisanpassungen für Software zum 01.01.2024. Online verfügbar unter https://www.cursor-distribution.de/de/home-de/community-informix/informix-news-details/1386-ibm-preisanpassungen-software-1-jan-2024, zuletzt geprüft am 02.03.2025.

CURSOR Software AG (2024): IBM Preisanpassungen für Software zum 01.01.2025. Online verfügbar unter https://www.cursor-distribution.de/de/home-de/community-informix/informix-news-details/1450-ibm-preisanpassungen-software-1-jan-2025, zuletzt geprüft am 02.03.2025.

Datanyze (2024): Marktanteile der führenden Unternehmen für Video- und Audiokonferenzsysteme. (Stand 25. September 2024). Statista Research Department. Online verfügbar unter https://de.statista.com/statistik/daten/studie/1228015/umfrage/marktanteile-der-fuehrenden-unternehmen-fuer-video-und-audiokonferenzsysteme/, zuletzt geprüft am 04.11.2024.

Diedrich, Oliver (2007): SugarCRM wechselt zur GPLv3. In: *heise online*, 26.07.2007. Online verfügbar unter https://heise.de/-155694, zuletzt geprüft am 04.11.2024.

Eberl, Markus (2016): Shorter Smarter Surveys. In: Bernhard Keller, Hans-Werner Klein und Stefan Tuschl (Hg.): Marktforschung der Zukunft - Mensch oder Maschine? Wiesbaden: Springer Fachmedien Wiesbaden, S. 217–230.

Föhl, Ulrich; Friedrich, Christine (2022): Quick Guide Onlinefragebogen. Wiesbaden: Springer Fachmedien Wiesbaden.

Free Software Foundation Europe (o.J.): Frequently Asked Questions on Free Software Licensing. Online verfügbar unter https://fsfe.org/freesoftware/legal/faq.de.html, zuletzt geprüft am 04.11.2024.

Frohmann, Frank (2022): Digitales Pricing. Wiesbaden: Springer Fachmedien Wiesbaden.

FSF; GNU (2001): What is Free Software? GNU. Online verfügbar unter https://www.gnu.org/philosophy/free-sw.en.html, zuletzt aktualisiert am 01.01.2024, zuletzt geprüft am 04.11.2024.

Gentemann, Lukas; Termer, Frank (2020): Open Source Monitor. Studienbericht 2019. Hg. v. Bitkom e.V.

Gentemann, Lukas; Termer, Frank; Weber, Anja (2021): Open-Source-Monitor. Studienbericht 2021. Berlin. Online verfügbar unter https://www.bitkom.org/sites/main/files/2021-12/211207-bitkom-studie-openmonitor-2021.pdf, zuletzt geprüft am 04.11.2024.

Goldacker, Gabriele (2017): Digitale Souveränität. 1. Aufl. Berlin. Online verfügbar unter https://www.oeffentliche-it.de/documents/10181/14412/Digitale+Souver%C3%A4nit%C3%A4t, zuletzt geprüft am 04.11.2024.

Hering, Ekbert (2014): Einfache Zuschlagskalkulation. In: Ekbert Hering (Hg.): Kalkulation für Ingenieure. Wiesbaden: Springer Fachmedien Wiesbaden (essentials), S. 3–7.

IEEE Computer Society (1994): IEEE standard for a software quality metrics methodology (IEEE Std 1061-1998). Online verfügbar unter https://ieeexplore.ieee.org/servlet/opac?punumber=6061.

Intevation GmbH (2024): Beantworteter Fragenkatalog zu OpenSlides. Osnabrück, 26.09.2024. E-Mail an Maximilian Overkamp.

Kirchgeorg, Manfred (2018): Kundenzufriedenheit. Online verfügbar unter https://wirtschaftslexikon.gabler.de/definition/kundenzufriedenheit-39738/version-263140, zuletzt aktualisiert am 15.02.2018, zuletzt geprüft am 16.02.2025.

Kollmann, Tobias (2018): Definition: Freemium. In: *Springer Fachmedien Wiesbaden GmbH*, 19.02.2018. Online verfügbar unter https://wirtschaftslexikon.gabler.de/definition/freemium-53522/version-276605, zuletzt geprüft am 22.10.2024.

Kompetenzzentrum Öffentliche IT (2019): Digitale Souveränität - Was brauchen wir zur staatlichen Selbstbestimmung im Digitalen? Digitalpolitisches Dossier #2. Mittwoch, 27. November 2019 Deutscher Bundestag. Online verfügbar unter https://www.oeffentliche-it.de/veranstaltungen/digitale-souveranitat, zuletzt geprüft am 04.11.2024.

KPMG; Bitkom Research (2022): Nutzung von Cloud Computing in Unternehmen in Deutschland in den Jahren 2011 bis 2022. Hg. v. Statista. Online verfügbar unter https://de.statista.com/statistik/daten/studie/177484/umf

rage/einsatz-von-cloud-computing-in-deutschen-unternehmen-2011/, zuletzt geprüft am 22.10.2024.

Krempl, Stefan (2022): BSI soll unabhängig, die Verwaltung mit Open Source souveräner werden. In: *heise online*, 20.05.2022. Online verfügbar unter https://heise.de/-7100848, zuletzt geprüft am 04.11.2024.

Kult, Wolfgang (o.J.): Zuschlagskalkulation / 2 Summarische Zuschlagskalkulation. Haufe. Online verfügbar unter https://www.haufe.de/finance/haufe-finance-office-premium/zuschlagskalkulation-2-summarische-zuschlagskalkulation_idesk_PI20354_HI2679500.html, zuletzt geprüft am 15.02.2025.

Lehmann, Sonja; Buxmann, Peter (2009): Preisstrategien von Softwareanbietern. In: *Wirtschaftsinformatik*, S. 519–529.

Lehmann, Sonja; Draisbach, Tobias; Koll, Corinna; Buxmann, Peter; Diefenbach, Heiner (2010): SaaS-Preisgestaltung: Bestehende Preismodelle im Überblick. In: Alexander Benlian (Hg.): Software-as-a-Service. Anbieterstrategien, Kundenbedürfnisse und Wertschöpfungsstrukturen. 1. Aufl. Wiesbaden: Gabler, S. 155–169.

Mecke, Ingo (2018): Gentlemen's Agreement. Gabler Wirtschaftslexikon. Online verfügbar unter https://wirtschaftslexikon.gabler.de/definition/gentlemens-agreement-33977/version-257493, zuletzt aktualisiert am 19.02.2018, zuletzt geprüft am 04.11.2024.

Mell, P. M.; Grance, T. (2011): The NIST definition of cloud computing. Gaithersburg, MD.

Microsoft Corporation (2022): Commercial Licensing Guide. January 2022. Online verfügbar unter https://www.microsoft.com/en-us/download/details.aspx?id=11091, zuletzt geprüft am 22.10.2024.

Microsoft Corporation (30.06.2024): Earnings Release FY24 Q4. Redmond, Wash. Online verfügbar unter https://www.microsoft.com/en-us/investor/earnings/FY-2024-Q4/press-release-webcast.

Mittermeier, Alexander (2022): Die größten Softwareunternehmen weltweit nach Umsatz im Jahr 2021. (in Milliarden US-Dollar). Statista; GeVestor. Online verfügbar unter https://de.statista.com/statistik/daten/studie/151056/umfrage/umsatz-fuehrender-software-hersteller-durch-software-in-europa/, zuletzt geprüft am 22.10.2024.

Müller, Steffen; Heim, Nina; Matthys, Stefan (2022): Was sind Kunden zu zahlen bereit? Ein Vergleich der Open-Ended-, Gabor-Granger- und Van-Westendorp-Methode. In: *Marketing Review St. Gallen* Januar 2022, 2022 (1), S. 10–15.

Open Source Iniative (2006): History of the OSI. Online verfügbar unter https://opensource.org/history, zuletzt aktualisiert am 31.10.2018, zuletzt geprüft am 04.11.2024.

Open Source Initiative (2006): The Open Source Definition (Annotated). Online verfügbar unter https://opensource.org/definition-annotated, zuletzt aktualisiert am 16.02.2024, zuletzt geprüft am 04.11.2024.

OpenSlides-Team (o. J.a): Elektronische Stimmabgabe. Online verfügbar unter https://openslides.com/de/elektronische-stimmabgabe/, zuletzt geprüft am 04.11.2024.

OpenSlides-Team (o. J.b): Referenzen. Online verfügbar unter https://openslides.com/de/referenzen/, zuletzt geprüft am 04.11.2024.

OpenSlides-Team (o. J.c): Video livestream. Online verfügbar unter https://openslides.com/en/video-live-stream/, zuletzt aktualisiert am 04.11.2024.

Oracle (o. J.a): About MySQL. Online verfügbar unter https://www.mysql.com/about/, zuletzt geprüft am 04.11.2024.

Oracle (o. J.b): MySQL as an Embedded Database. Online verfügbar unter https://www.mysql.com/de/oem/, zuletzt geprüft am 04.11.2024.

Oracle (o. J.c): MySQL Technical Support. Online verfügbar unter https://www.mysql.com/support/, zuletzt geprüft am 04.11.2024.

Oracle (o. J.d): Oracle MySQL. Online verfügbar unter https://shop.oracle.com/apex/product?p1=MySQL, zuletzt geprüft am 04.11.2024.

Oracle (2024a): MySQL Community Downloads. Online verfügbar unter https://dev.mysql.com/downloads/, zuletzt geprüft am 04.11.2024.

Oracle (2024b): Oracle Software Technical Support Policies. Effective Date: 05-January-2024.

Oram, Clint (2014): SugarCRM in the Next 10 Years. Online verfügbar unter https://web.archive.org/web/20160224192704/https://community.sugarcrm.com/thread/18434, zuletzt geprüft am 04.11.2024.

Oram, Clint (2018): Sugar Community Edition open source project ends. Online verfügbar unter https://sugarclub.sugarcrm.com/engage/b/sugar-news/posts/sugar-community-edition-open-source-project-ends, zuletzt geprüft am 04.11.2024.

Oslak, Peter (2023): Das kann Dynamic Pricing. In: *Industriemagazin*, 05.07.2023 (Nr. 07-08), S. 78–80. Online verfügbar unter https://www.wiso-net.de/document/OEIM__d1b282a2c6609ab72b1c881851bce8cb807828bc, zuletzt geprüft am 20.02.2025.

Porter, Michael E. (1991): Towards a dynamic theory of strategy. In: *Strategic management journal*.

Priester, Anna (2022): Dynamic Pricing aus Konsumentensicht. Wiesbaden: Springer Fachmedien Wiesbaden.

Raithel, Jürgen (2008): Grundlagen und -probleme empirischer Sozialforschung. In: Jürgen Raithel (Hg.): Quantitative Forschung. Wiesbaden: VS Verlag für Sozialwissenschaften, S. 11–23.

Red Hat (25.03.2019): Red Hat Reports Fourth Quarter and Fiscal Year 2019 Results. Raleigh, North Carolina. Online verfügbar unter https://www.redhat.com/de/about/press-releases/red-hat-reports-fourth-quarter-and-fiscal-year-2019-results, zuletzt geprüft am 04.11.2024.

Reinders, Heinz (2011): Fragebogen. In: Heinz Reinders, Hartmut Ditton, Cornelia Gräsel und Burkhard Gniewosz (Hg.): Empirische Bildungsforschung. Wiesbaden: VS Verlag für Sozialwissenschaften, S. 53–65.

Riehle, Dirk (2007): The Economic Motivation of Open Source Software: Stakeholder Perspectives. In: *Computer* 40 (4), S. 25–32. DOI: 10.1109/MC.2007.147.

Schäfers, Björn (2004): Grundlagen der Zahlungsbereitschaft und Methoden ihrer Messung. In: Björn Schäfers (Hg.): Preisgebote im Internet. Wiesbaden: Deutscher Universitätsverlag, S. 9–51.

Schnaak, Greta; Termer, Frank (2023): Open-Source-Monitor. Studienbericht 2023. Berlin. Online verfügbar unter https://www.bitkom.org/sites/main/files/2023-09/bitkom-studie-open-source-monitor-2023.pdf, zuletzt geprüft am 04.11.2024.

Sehgal, Naresh Kumar; Bhatt, Pramod Chandra P. (2018): Cloud Computing. Cham: Springer International Publishing.

Shahrivar, Shahrokh; Elahi, Shaban; Hassanzadeh, Alireza; Montazer, Gholamali (2018): A business model for commercial open source software: A systematic literature review. In: *Information and Software Technology* 103, S. 202–214. DOI: 10.1016/j.infsof.2018.06.018.

Shapiro, Carl; Varian, Hal R. (1999): Information rules. A strategic guide to the network economy. Boston, Mass.: Harvard Business School Press. Online verfügbar unter http://bvbr.bib-bvb.de:8991/F?func=service&doc_library=BVB01&doc_number=008441201&line_number=0002&func_code=DB_RECORDS&service_type=MEDIA.

Spasojevic, Anastazija (2024): What Is a Line of Code (LOC)? phoenixNAP. Online verfügbar unter https://phoenixnap.com/glossary/line-of-code-loc, zuletzt aktualisiert am 05.12.2024, zuletzt geprüft am 15.02.2025.

Stallman, Richard M. (1986): What is the Free Software Foundation. Gnu's Bulletin. GNU (1). Online verfügbar unter https://www.gnu.org/bulletins/bull1.txt, zuletzt geprüft am 04.11.2024.

Stanciu, Alin-Marius; Ciocârlie, Horia; Julean, Carla-Patricia (2023): Electronic Voting System Based on the Blockchain Technology. In: 2023 International Conference on Electrical, Computer and Energy Technologies (ICECET). 2023 International Conference on Electrical, Computer and Energy Technologies (ICECET). Cape Town, South Africa, 16.11.2023 - 17.11.2023: IEEE, S. 1–6.

Statistisches Bundesamt (2023): Nutzung von Cloud Computing nach Beschäftigtengrößenklassen. Statistisches Bundesamt. Online verfügbar unter https://www.destatis.de/DE/Themen/Branchen-Unternehmen/Unternehmen/IKT-in-Unternehmen-IKT-Branche/Tabellen/iktu-06-cloud-computing.html, zuletzt geprüft am 22.10.2024.

Stefanovic, Darko; Nikolic, Danilo; Dakic, Dusanka; Spasojevic, Ivana; Ristic, Sonja (2020): Static Code Analysis Tools: A Systematic Literature Review. In: Branko Katalinic (Hg.): Proceedings of the 31st International DAAAM Symposium 2020, Bd. 1: DAAAM International Vienna (DAAAM Proceedings), S. 565–573.

SugarCRM (o. J.): Startseite. Online verfügbar unter https://www.sugarcrm.com/de/, zuletzt geprüft am 04.11.2024.

SugarCRM (2011a): Go Pro. Online verfügbar unter https://web.archive.org/web/20110501022904/http://www.sugarcrm.com/crm/gopro/gopro.html, zuletzt aktualisiert am 01.05.2011, zuletzt geprüft am 04.11.2024.

SugarCRM (2011b): Sugar Editions & Pricing. Online verfügbar unter https://web.archive.org/web/20110501022732/http://www.sugarcrm.com/crm/products/editions.html, zuletzt aktualisiert am 01.05.2011, zuletzt geprüft am 04.11.2024.

SuiteCRM (2017): SugarCRM Community Edition EOL doesn't mean the end of your CRM. Online verfügbar unter https://suitecrm.com/sugarcrm-community-edition-eol-doesnt-mean-the-end-of-your-crm/, zuletzt geprüft am 04.11.2024.

Urban & Vogel (2015): Ein Software-Wechsel muss sich wirklich lohnen. In: *MMW Fortschritte der Medizin* 157 (8), S. 14. DOI: 10.1007/s15006-015-3007-4.

Woo (o. J.a): Enterprise ecommerce that scales. Grow with Woo Enterprise. Online verfügbar unter https://woocommerce.com/enterprise-ecommerce/, zuletzt geprüft am 04.11.2024.

Woo (o. J.b): Getting Started on the Woo Marketplace. Online verfügbar unter https://woocommerce.com/document/marketplace-overview/, zuletzt geprüft am 04.11.2024.

Woo (o. J.c): Partners. Expand your business with the WooCommerce Marketplace. Online verfügbar unter https://woocommerce.com/de/partners/, zuletzt geprüft am 04.11.2024.

Woo (o. J.d): Trusted hosting for your WooCommerce store. Online verfügbar unter https://woocommerce.com/de/hosting-solutions/, zuletzt geprüft am 04.11.2024.

Woo (o. J.e): What is WooCommerce? Online verfügbar unter https://woocommerce.com/woocommerce/, zuletzt geprüft am 04.11.2024.

Woo (o.J.f): WooCommerce Brands. Installation. Online verfügbar unter https://woocommerce.com/document/woocommerce-brands/, zuletzt aktualisiert am 04.11.2024.

Woo (o. J.g): WooCommerce Development Services. Get expert help creating the store of your dreams. Online verfügbar unter https://woocommerce.com/development-services/, zuletzt geprüft am 04.11.2024.

Woo (o. J.h): WooCommerce-Erweiterungen. Online verfügbar unter https://woocommerce.com/de/product-category/woocommerce-extensions, zuletzt geprüft am 04.11.2024.

Woo (o. J.i): WooCommerce-Erweiterungen. Collections. Online verfügbar unter https://woocommerce.com/de/collections/, zuletzt geprüft am 04.11.2024.

Woo (2019): Product Recommendations for WooCommerce. Online verfügbar unter https://woocommerce.com/de/products/product-recommendations/, zuletzt geprüft am 04.11.2024.

Woock, Kristina; Meinert, Nele; Völtzer, Linda; Nordholt, Paul; Busch, Susanne (2022): Nutzwertanalyse: Optionen systematisch bewerten. In: *Pflegez* 75 (6), S. 16–19. DOI: 10.1007/s41906-022-1254-4.

Development of a Priority Rule Based Traffic Management Policy for Standardized AGV Conflict Zone Coordination

Jeffrey Feufel
Hochschule Pforzheim
Business School
Tiefenbronner Str. 65
75175 Pforzheim
feufelje@hs-pforzheim.de

Frank Schätter
Hochschule Pforzheim
Business School
Tiefenbronner Str. 65
75175 Pforzheim
frank.schätter@hs-pforzheim.de

Julian Popp
MHP Management- und IT
Beratung Gmbh
Digital Factory & Supply Chain
Hindenburger Str. 45
71638 Ludwigsburg
julian.popp@mhp.com

Key Words
Automated Guided Vehicles, Traffic Management, Conflict Zones, Heterogenous Fleets, VDA5050

ABSTRACT (English)
The use of Automated Guided Vehicles (AGVs) has proven to be effective in companies across industries. In intralogistics, AGV fleets are constantly growing, making their coordination a challenge, especially in conflict areas such as junctions. Therefore, this paper presents a novel traffic management policy that promises reliable coordination of AGVs in dynamic traffic situations through the use of prioritisation rules in user-specific scenarios. The focus is on improving the adaptability and scalability of the often rigid coordination methods in current control systems, by providing scheduling templates in order to prepare them for increasing coordination requirements. The contribution thus provides a basis for the further development of the application of standardised control systems and contributes to the sustainable optimization of automated material flows.

ABSTRACT (German)
Der Einsatz von fahrerlosen Transportsystemen, sogenannten Automated Guided Vehicles (AGVs), hat sich in Unternehmen, insbesondere in der Intralogistik, als effektiv erwiesen. Da die Zahl der AGV-Flotten stetig wächst, wird ihre Koordination, insbesondere in Konfliktbereichen wie Kreuzungen, zunehmend herausfordernder. In diesem Beitrag wird daher eine neuartige Verkehrsmanagementrichtlinie vorgestellt, die eine zuverlässige Koordination von AGVs in dynamischen Verkehrssituationen durch die Verwendung von Priorisierungsregeln in benutzerspezifischen Szenarien verspricht. Der Schwerpunkt liegt auf der Verbesserung der Anpassungsfähigkeit und Skalierbarkeit der oft starren Koordinationsmethoden aktueller Steuerungssysteme durch Bereitstellung von Planungsvorlagen, um sie auf steigende Koordinationsanforderungen vorzubereiten. Der Beitrag liefert somit eine Grundlage für die Weiterentwicklung der Anwendung standardisierter Steuerungssysteme und trägt zur nachhaltigen Optimierung automatisierter Materialflüsse bei.

1. Introduction

It is evident that companies have recognised the central role of technological enablers such as robotics, in the realization of the Industry 4.0 vision (Deloitte 2023). As a result, solutions that lead to the digitalisation and automation of operational processes are taking a central position within organizational hierarchies (Kagermann et. al 2023). Automated Guided Vehicles (AGVs) are a key tool in this context, particularly within in production and logistics. This is because AGVs facilitate the execution of transport processes with increased efficiency and reduced operating costs compared to traditional material flow approaches (Fottner et al. 2022, Nikelowski & Wolny 2020).
However, the implementation of automation projects to integrate AGVs is progressing at a slow pace (Aguiar et al., 2019). This is due to the high investment costs and limited budgets of users, which prevent the implementation of transformation plans all at once. It is common for new product innovations to appear on the market between projects. As a result, heterogeneous vehicle fleets emerge with each new tender. To make matters even worse, AGV manufacturers are often reluctant to cooperate between their technologies, resulting in system incompatibilities that make communication and coordination between material handling systems difficult. The challenge currently facing companies is to integrate multiple, usually proprietary, control systems of AGVs from different manufacturers into the higher-level control architecture in order to ensure error-free operations (Ullrich & Albrecht 2023). This leads to more inefficiencies instead of the planned gains that should result from the proposed synergy effects.
To solve this problem and exploit the potential of automated material flow in the future, the VDA-5050 guideline has been developed in Germany by the Association of German Automobile Manufacturers starting in 2019. This guideline provides a basis for companies to implement specialised solutions for the standardised control of AGVs on the basis of MQTT-communication interfaces. By now the first-come, first-served (FCFS) principle is now widely used for traffic management due to its low complexity character (Nils 2022). This means that AGVs are sorted according to their arrival time at junctions. It

is also important to note that the coordination is based on only one decision parameter (arrival time at junctions) and does not take into account priorities or queues when determining the right of way.
As fleet sizes increase, the complexity of traffic management based on a centrally controlled system with one decision parameter is likely to become unmanageable. As the coordination effort increases with each additional vehicle, it is necessary to analyse whether current coordination methods can continue to meet the requirements of efficiency and safety. Especially at high vehicle densities, there is a risk that the FCFS principle will reach its limits in terms of the efficient coordination of all vehicles especially in terms of efficient throughput.
Think of the presence of AGVs with orders that have different priorities, or sections of the layout with higher traffic volume and therefore longer queues. It is therefore essential to explore ways of improving the control system to ensure robust AGV coordination in the future. This is because the challenges posed by high traffic density can be applied to intralogistics systems by referring to scenarios from the real world of transport, where a higher risk of disruption at junctions is expected as the size of the system increases. Avoiding congestion due to bottlenecks or deadlocks is crucial. In particular, disruptions to the flow of traffic have the potential to cause delays in throughput times, which in turn can set off a detrimental chain reaction with the potential to spread throughout the system.
The risks associated with such disruptions highlight the challenges of coordinating future fleets based on the current 'first come, first served' (FCFS) principle. The inability to compare and prioritise AGVs based on their order information is a problem due to the limited adaptability in coordinating heterogeneous fleets. Consequently, there is a need to extend traffic management with a novel coordination methodology that is capable of coordinating conflict areas of a traffic network not only from a safety point of view, but also with increased efficiency in the long run. Therefore, the aim of this paper is to design traffic rules that specifically exploit the intersection control for AGVs under consideration of several different decision parameters. Special attention is paid to the integration of these rules into a policy and then into an automated decision process, which can be implemented in existing control architectures or systems with little effort.
The remainder of the paper is structured as follows. In the next section, we introduce and summarise the evolution of AGVs for logistics automation and traffic management methods, highlighting current challenges and approaches. We then present the design of the new traffic policy and its integration into an automated decision making process based on the functionality of the VDA5050 guideline. The result of this process is the formulation of four novel priority rule-based scheduling approaches that serve as templates for specialised combinations. The extensions are implemented in the existing control infrastructure through the configuration of an interactive auction process (intersection as a market), which is based on the autonomous intersection management approach (AIM) by Dresner and Stone in 2006 and its extension, the platoon-based intersection management approach (PAIM) by Bashiri and Flemming from 2017. Finally, a first validation is performed and our research is summarised and directions for future research are highlighted.

2. Logistics automation and traffic management

Firstly, the functionality of driverless transport systems is explained. In addition, a review of the existing literature has been undertaken to identify the current state of the art in traffic management concepts and the prevailing challenges in this area.

2.1 Driverless transport systems

AGV systems can be categorised as automated material flow systems. They are considered to be floor-bound discontinuous conveyors that are used to transport goods from a source to a sink (Scholz 2019, Müller 2011). The term therefore covers scenarios of goods transport processes in production and warehouses that are realised by automated floor-bound vehicles (Ullrich and Albrecht 2023). According to the VDI 2510 guideline, AGVs are: '*floor-bound systems that can be used inside and/or outside buildings. They essentially consist of one or more automatically controlled, contactless guided vehicles with their own traction drive and, if required, of a) a guidance control system, b) equipment for determining location and position detection, c) data transmission equipment and d) infrastructure and peripheral equipment.'* AGVs are the operational workers of the internal transport system. Due to their variety of applications, they can differ in their mode of operation (structure, payload) and their degree of automation (sensor technology, communication, decision-making ability) (Ullrich and Albrecht 2019). It is therefore necessary to consider two categories of vehicles separately, firstly AGVs, and *A*utonomous *M*obile *R*obots (AMRs). The latter can navigate freely through sophisticated software modules and thus offer greater flexibility in dynamic material flow situations (Fragapane et al. 2021).
Every AGV includes navigation technology and a guidance system for master control (Pichler 2011; Schwarz et al. 2013).
Navigation can be seen as the 'eyes' of the system, with AGVs being unable to orient themselves in unfamiliar environments (Kubasakova et al. 2024). They also lack the ability to make independent decisions about braking, acceleration or other actions. Rather, they use sensors to detect whether they are on the correct path of their transport route or deviating from it (Ullrich & Albrecht 2023). This limitation highlights the need for two coordinate systems to enable navigation, which can refer to a stationary layout and a reference system located in the centre of the vehicle (Conette 2013). The result of this dual-based method is a map with x, y and z coordinates

through which the vehicle navigates. Guidance and localisation methods are then used for operational control, with position determined by measuring wheel revolutions (odometry) (Pichler 2011). In addition, bearing is used to periodically interrogate the position using passive or active localisation technologies such as markers or line guidance (Hertzberg et al. 2012).
The *master controller* symbolises the "brain" of the driverless transport system (Dickmann et al. 2015). Its functional modules integrate the AGVs into the operational transport system, it serves as an interface between the clients and the operational vehicle level (Scholz 2019). Clients are either manual users or the internal material flow control system, which automatically generates transport orders. As soon as these are received, the transport order processing is activated. Firstly, all orders are grouped together and organised hierarchically as part of order management. If there is an order priority, this ensures that all orders can be allocated on time and in accordance with requirements. This also involves finding a suitable AGV and its optimal route. Taking into account the database, route planning algorithms simulate the route from the start to the end point (Dilefeld 2023). As a result, a free AGV can be tasked with the execution via vehicle scheduling. Furthermore, the information flow of the operation is not only downstream, but also upstream, as the vehicles report the order status via defined communication protocols. Ultimately, the control system links the host systems Enterprise Resource Planning (ERP), Warehouse Management System (WMS), Production Planning and Control System (PPC) and Internal Transport System (ITS) with the operational AGVs, thus enabling cross-system processes to be handled.

2.2 Traffic management methods

Traffic management concepts include the modelling of the overall environment as well as traffic coordination and its scheduling principles. Despite their different approaches, management concepts share a common goal. Coordination aims to maximise the efficiency of the overall system (Le Anh 2005). To achieve this goal, it is necessary to minimize bottlenecks and eliminate deadlocks and collisions. Three main concepts can be derived from the literature, which can be categorised as fully centralised, partially centralised and decentralised (Nils 2022). It should be noted, however, that these concepts do not have static boundaries. Rather, it is possible to combine functional modules of different procedures, so that hybrid solutions are possible (Fottner et al. 2021). The evaluation of the performance of a traffic control concept is based in particular on the coordination of conflict areas. These are perceived as constraints or bottlenecks in the system. Conflict areas are therefore characterised by an overlap of at least two AGV routes (Braun-Schweiger 2017).
In the following sections, the state of the art of traffic management systems is discussed, coordination with rule-based approaches is introduced, and challenges are highlighted.

2.2.1 State of the Art

There are two ways to implement a fully centralised approach (Nils 2022). In the query-based approach, AGVs transmit their planned route to the control centre when they wish to pass and wait for approval from the central control (Dresner and Stone 2008). There is also the assignment-based approach, where the central control instance transmits trajectories in the form of time-window-based assignments to AGVs in the detection zone (Yang et al. 2016). Although this requires more planning effort, the assignment allows the integration of further control mechanisms such as priority rules (Khayatian et al. 2020).
Furthermore, in the partially centralised approach, AGVs act autonomously in individual steps. In particular, route planning is carried out by the vehicle itself (Nils 2022). This concept often involves a combination of centralised and decentralised components, which is precisely why experts see great potential in it, especially in terms of improved scalability compared to fully centralised approaches (Qian et al. 2017). Unlike the two centralised approaches, the decentralised approach does not require a central control system. Vehicles operate in multi-agent systems (MAS) and coordinate themselves according to their planned routes (Schaffer and Weidenbach 2019). This form of coordination often takes place using token or auction procedures (Carlino et al. 2013). In areas of conflict, AGVs then communicate via pre-defined negotiation protocols. In this context, we can also speak of a cooperative control concept (Basile et al. 2019) which some companies describe this as swarm based.
Throughout the literature review, there is disagreement as to which of the three concepts has the highest system/coordination efficiency. Depending on the scenario, conflicting claims are made, with Pratissoli et al. arguing that centralised control systems generally have an advantage over decentralised systems (Pratissoli et al. 2023). In contrast, Fragapane et al. argue that large vehicle fleets in particular cannot be coordinated efficiently by centralised entities. In general, the results of application-specific simulations of research projects should be treated with caution. This is because central instances have access to global information and thus encompass the entire intralogistics system, whereas decentralised structures mainly use local information for the coordination decision (Fragapane et al. 2021). From this it can be concluded that centralised instances in complex systems do indeed have statistically higher processing times (Schmidt et al. 2020). However, this does not mean that they are less efficient than decentralised approaches in general. The centralised instance searches for the maximum performance of each vehicle and takes into account all possible points of conflict (Siegfried and Bourafa 2023). Decentralised concepts are based on local information, so it may happen that a new route solution has to be found for a vehicle at every conflict point along its route from source to sink (Preisler 2016). This leads to the conclusion that although sub-problems can be solved more quickly with decentralised methods, their overall

processing time in complex traffic situations may be longer than with a centralised solution.
However, research agrees that centralised approaches make the system less robust against failures (vulnerability to failure) (DeRyck et al. 2020). This is related to the fact that the coordination effort for a higher-level instance usually becomes too high with increasing system size (Günthner et al. 2012). In general, the functioning of a centralised lead authority is fundamentally opposed to the idea of autonomy, which proposes flexibility (Fottner et al. 2011). However, centralised concepts still have more areas of application than decentralised structures, as MAS are not yet sufficiently mature (Pratissoli et al. 2023). Although the potential of decentralised processes in terms of increasing flexibility and scalability is evident from practical applications, the development step from research to widespread application has not yet taken place (Schreiber 2013). However, it is expected that this will change in the coming years due to increasing complexity and ongoing research in this area.

2.2.2 Coordination with rule based approaches

When intralogistics traffic control systems are based on rule-based approaches, FCFS is the most common method in practice (Nils 2022). If a conflict area, e.g. an intersection, is currently occupied by a vehicle, each additional arriving vehicle sends a request to the traffic control system to pass through. This results in a time-ordered queue. This queue is then processed in order of registration time ('first in first out' (FIFO)). It should also be noted that other control concepts, better known as scheduling policies, have been established in real traffic scenarios over the years. These are usually tailored to the needs of the particular transport system and often use more than one decision parameter for the coordination decision (Nils 2022). Most designs are based on rights of way, which can be enforced on the basis of priorities and are implemented in the form of sorting procedures of requests within the central intersection control system (Guney and Raptis 2020). In this way, decision parameters can be defined depending on the focus of the performance orientation.
The AIM (Autonomous Intersection Management) project by Dresner and Stone in 2006 paved the way for this type of planning at conflict areas. The results of the project were to control autonomous vehicles (AVs) at real intersections not only using the FCFS principle, but also taking into account priority classes and making appropriate adaptations to traffic signal control for transit (Dresner and Stone 2008). As a result, prioritised vehicles achieved better throughput times, reduced delays and better on-time performance. Building on these findings, Bashiri and Flemming extended the AIM approach by considering whole groups from the same intersection entries. The PAIM (Platoon Based Autonomous Intersection Management) approach achieved lower average waiting times compared to the FCFS approach (Bashiri et al. 2017).

It can be seen that the coordination of traffic flow is often achieved by a combination of different control concepts. A skilful combination promises a positive effect on the performance indicators of the overall system (Nils 2022). Based on the results of the literature overview, a partially centralised approach seems to be the best choice for the current level of automation. Thus, a hybrid approach is being developed that combines the strengths of decentralised and centralised approaches and attempts to mitigate their weaknesses.

2.2.3 Challenges of traffic flow

In the context of AGV coordination, there are three main traffic flow challenges to be considered, namely *collisions*, *congestion* and *deadlocks*. These lead to interruptions in the traffic flow, which in turn affect the performance of the system (Fottner et al. 2022). In practice, there is a risk of collisions between AGVs and obstacles. Collisions can occur between two road users vehicle-to-vehicle (V2V) or collisions with an obstacle in the driving environment vehicle-to-obstacle (V2O) (Dharmasiri et al. 2019). Collisions of any kind are the cause of congestion, as temporary reductions in capacity create a bottleneck in the system (Zheng et. al, 2019; Deutscher Bundestag 2020).
Another cause of congestion can be excessive traffic demand in relation to the available infrastructure (Deutscher Bundestag 2020). These natural bottlenecks are often located at conflict points, resulting in low average speeds, longer waiting times and longer throughput times for road users (Strohhäussl 2007). As the proportion of self-driving or automated vehicles increases, so does the risk of congestion. This phenomenon occurs when one or more competing processes in a system block each other because the resource requirements of the processes, for example the release of the road section ahead, can never be satisfied (Lu et al. 2021). However, a simple traffic jam does not imply a deadlock, because according to Coffman (1971) four conditions must be met. Firstly, there must be 'mutual exclusion', as the resource (route section) cannot be used by more than one vehicle according to the VDA5050 guideline. Secondly, AGVs must occupy resources that have already been reserved while waiting for others to be released ('hold and wait'). Thirdly, a deadlock requires that resources are held by vehicles until completion and cannot be released in any other way ('no preemption'). Fourth, tasks must form a chain so that each task waits for one or more resources held by the next task in the chain ('circular wait') (Coffman 1971).

3. Development of rule sets for the systematic coordination of conflict areas

Our aim is to enhance the FCFS control system by implementing rule-based scheduling decisions. These could ensure that both order priorities and other decision parameters can be taken into account in the future. This section describes the development of a priority rule-based

traffic management policy for standardised AGV conflict zone coordination. The focus is on the systematic derivation of priority rules aimed at efficiently resolving conflicts and ensuring smooth AGV operation at critical junctions.

3.1 Preliminary considerations

The basic system requirements for an adaptive control extension are already in place; we assume that the order priority and more detailed order information is already transferred from the control systems to the assigned AGV via the VDA5050 interface. However, due to the FCFS logic, this information is ultimately not used in the coordination process. In this way, a potential is lost that could presumably have a positive effect on the ability to react in dynamic traffic situations. To build on this idea, the following points are necessary to prepare the control extension of intersection management: i) AGVs operate in a network of nodes and edges, ii) the control extension is based on the functionality, notation and guidelines of the VDA5050 guideline version 2.0.0 (VDA 2019).
Therefore, if an AGV wants to drive along an edge that is part of a conflict area, this edge must first be free. A crossing is considered free if all its involved edges are neither reserved nor occupied, or if the maximum number of AGVs has not yet reached its threshold:The 'maxAGVCount' parameter limits the number of AGVs that can be in the crossing area at the same time (capacity regulation).
The intersection is not accessible if it is currently reserved for or occupied by another vehicle. If the conflict zone is currently inaccessible, the AGV will stop at the node defined in the configuration as a holding/last stop-point, because its released part of the route (base) can not be extended. Normally a holding point is set at the last node before the actual entry into the junction.
The intersection is released according to the control logic of the VDA5050 interface at the first node that is no longer included in the intersection area. This means that the release takes place n+1 nodes after the actual departure from the intersection. If the junction is free for the next AGV in the queue, all edges of the planned route (horizon) within the junction and as many edges as possible after the junction are reserved for the AGV and transferred to the vehicle-specific base. This process is repeated for each additional AGV.

3.2 Novel rule-based traffic management policy

The following sections describe the development of the novel rule-based traffic management policy. In section 3.2.1, general assumptions regarding the structure and application of the later rules are highlighted, while section 3.2.2 focuses on the auction process for traffic management. The core of the traffic management extension are the four new rule-based control approaches (templates) with their scheduling policies in section 3.2.3. Finally, a security protocol is added to the policy in section 3.2.4, primarily to integrate security mechanisms for operational purposes.

3.2.1 General assumptions

As long as there is only one AGV in a conflict area (intersection), or only one AGV wants to enter the conflict area, it can be assumed that the FCFS principle will continue to apply without any problems. The FCFS principle therefore remains as the basic control principle and fallback option. However, it will be replaced by the new set of rules as soon as a number of $n \geq 2$ AGVs are in the detection range of the conflict area and they do not reach the same access route. In order to decide which AGV attains the right of way first, an *auction process* with a decision protocol is needed. Further the idea is, that each AGV arriving at the intersection sends its order information to the master control, which then determines the rank in the dispatching list based on the active approach. Precisely this process integrates the mentioned decentralized component into the centralized master control and should allow us to gain the ability of adaptive scheduling in dynamic traffic scenarios. While taking several decision paramteres from each AGV into account, our hybrid control mechanism takes action of the auction between the vehicles. Firstly the implementation of the control approaches within an auction process requires a restructuring and extension of the map logic according to the VDA 5050 guideline. Figure 1 shows the new conflict area and its map elements. Compared to the original logic of the VDA5050-guideline the function of entry, exit and release nodes of the intersection does not change. In contrast the new additions are the triggers registration node and decision node. In the following the differences between the original VDA5050 logic and the extension will be explained in detail.

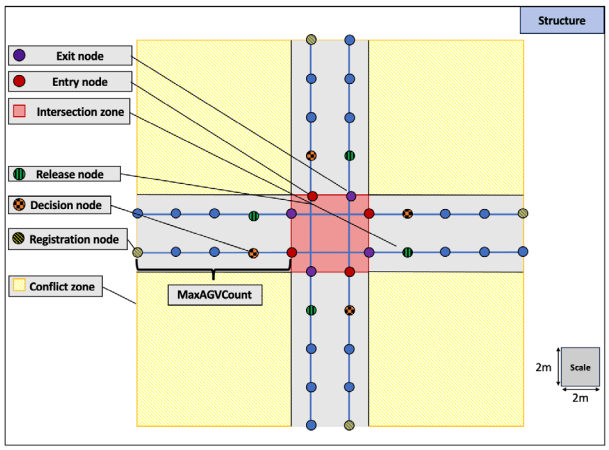


Figure 1: Illustration of the new conflict area

Detection range

Currently the VDA5050 logic is based on the idea, that the request and decision to extend the specific AGV-base is only made shortly or directly at the intersection entry node (Figure 1, red node). However, this request must be made in advance , both in terms of time and space, since

the sorting of transit requests from multiple AGV is implemented by an automated process, signals need to be exchanged and processed by the master control software. Conversely, this sorting process requires a time buffer for effective decision making. For this reason, the detection range is defined in such a way that it can guarantee a distance between the request of the AGV and its actual arrival at the intersection. It is therefore possible to include other AGVs transit requests in the same decision run. In summary, the registration node of a conflict area is always embedded n nodes in advance to the actual intersection entry node. The maxAVGCount for the intersection arm is then determined by the amount of nodes between and including the registration node and intersection entry node. As soon as the maxAGVCount reaches its threshold value at the intersection arm, it is considered occupied. If no space becomes available, no further vehicle can register for transit via this intersection arm (complexity reduction).
In practice, each conflict area can be adapted according to its requirement profile. It should be noted that the distance between the registration node and the intersection entry node affects the maxAGVCount variable. Depending on the distance between the two nodes, the maximum queue length per junction arm is determined. It is also possible that conflict areas have different access times, in which case an exception would have to be configured to use a control approach.

Decision node (trigger)

The decision node (Figure 1, orange-black patterned node) is the trigger for stopping the sorting procedure in the decision run. As soon as a vehicle reaches this marker, the recording of new transit requests in the respective decision run is temporarily stopped. The master controller must determine whether the AGV at the decision node can pass the conflict area directly or whether it must wait at the entrance to the intersection. Depending on the two options, either the junction and the maximum number of nodes behind the junction are reserved and transferred to the base of the vehicle. Or the horizon remains unchanged due to another vehicle with right of way and the AGV continues its route only to the end of its current base, which is normally the intersection entry node. From a functional point of view, the trigger should therefore be located in the immediate vicinity of the intersection entry node to ensure a sorting interval that is as long as possible to allow the best coordination decision. However, reaction and communication delays must also be taken into account when determining the exact position. In general, the decision node can theoretically be assigned to any node on the crossing arm. However, the interdependence between the length of the sorting path and the safety buffer for information transmission must be taken into account. As the decision point is moved closer to the crossing entrance, the buffer for the sorting process is reduced. The decision node with its trigger is assigned to the last node before the actual entrance node to the crossing (rule of thumb). The decision node interrupts the sorting process of the transit requests in the Crossing-Manager. The trigger function is activated as soon as an AGV reaches the decision node. The activated trigger function is ignored if the intersection is occupied when the AGV arrives at the logon node.

3.2.2 Interactive auction process

The master control is responsible for coordinating the intersection area. It collects and processes the transit requests of incoming AGVs. The processing is carried out by using a sorting process that sequences requests according to the active control approach, thereby creating a ranked scheduling list.
As a result of the request to the master control, the vehicles are initially assigned to the conflict area for organisational purposes until they have successfully passed through it. Within the conflict area, system rules adapted for AGVs apply (see Table 1).

Table 1: System rules

Rule	Description
(1)	AGVs reduce or increase their speed (v) to 1 m/s and maintain it until they reach the decision node.
(2)	AGVs can only travel to the next node when the AGV in front of them has reached the node after next (collision prevention).

Each AGV registers its transit request with the following information in order to be included in the sorting process: agvId (vehicle identification number), orderId (order identification number), timestamp (time stamp), priorityvalue (priority value), requestnodeId (identification number of the registration node), orderdeadline (deadline) and speed value (transit time).
Within the request list, the crossing sequence is updated during the sorting process in the decision run with each newly added request. If the first AGV triggers the decision node, the sorting freezes and the vehicle at the first queue position is given clearance to pass. The sorting runs and their interruptions are now referred to as hot and frozen gaps:

- *Hot gap*: Refers to the time interval for sorting new or existing transit requests within a decision run.
- *Frozen Gap*: This refers to the time interval during which the sequence is frozen and no new requests can be sorted.

As soon as the AGV at the first list position has received the transit clearance and passes the intersection accordingly, the interruption can be cancelled. Since the intersection is occupied at this point in time, requests can be considered again for the next decision run. At this point in time, no other AGV is able to pass the intersection area according to the deadlock safety logic. The second hot gap can be maintained until the AGV (or the last vehicle of a platoon) reaches the intersection exit node. Then the sorting is closed again and the decision run comes to the conclusion which AGV/group is allowed to pass the intersection area next. This results in a cycle that repeats itself as long as there are $n \geq 2$ requests from different

request nodes in the list, until all AGVs waiting to pass the intersection have been processed.
The following rules apply to the hot and frozen gaps. The starting point is an empty conflict area.

Table 2: Hot gap and frozen gap rules

Rule	Description
(1)	The first hot gap extends over the time interval between an AGV registering and reaching the decision node.
(2)	The AGV with the earliest transit request also reaches the decision node first, in accordance with system rule (1) (table 1).
(3)	The arrival of the AGV at the decision node interrupts the sorting process and is considered the catalyst for the start of the frozen gap (transition to the frozen gap).
(4)	Frozen Gap 1 lasts until the AGV with the travel clearance passes the entrance node of the intersection, thus enabling the next hot gap (transition phase).
(5)	The next decision run takes place while the passing AGV is between the entry and exit nodes of the intersection.
(6)	Frozen Gap 2 interrupts the sorting process again until the next AGV (n+1) reaches the intersection entry node.

It should be noted that AGVs can still register during the frozen gap period if the maxAGVCount threshold for one of the junction arms has not yet been reached. However, the transit request will not be considered until the next decision run (hot gap).

3.2.3 Prioriry rules

This section presents the four novel scheduling rules for heterogenous fleets at intralogistics intersections. Each approach is build upon using cascadic decision protocols which try to enhance the overall efficiency of coordination at junctions while maintaining deadlock prevention and safety measures for operational workflow. Please note that these approaches serve as templates for further combinations and ongoing optimization. Table 2 includes an overview for all operators with their respective descriptions.

Table 2: „List of operators“

Variable	Naming and Description
D	*orderdeadline* – Remaining time until the order deadline (in seconds).
F	*VehicleId*- Unique identifier for an AGV in the system.
F*	*VehicleId* with the right of way (determined by the master control in the active decision run).
F_{max}	*VehicleId* with the highest priorityvalue (P) in a queue.
G_{active}	Speed orientation mode
K	*CrossingarmId*- Unique identifier for a specific intersection arm.
N	*GroupId* – Platoon of AGV sharing the same R and $T \leq T(F^*)$.
P	priorityvalue-priority of the order
Q	*QueueId* – Ordered list of AGV waiting at a crossing arm.
R	*RequestnodeId* – Unique identifier for a crossing arm (entry point of an AGV).
S	Speedvalue – Estimated transit time for an AGV
T	*Timetstamp* – Time of AGV transit request submission.
Z	Z_H = Zone assignment for the main road (high priority)
	Z_{NB} = Zone assignment for the side road (low priority)

First and foremost F* are not necessarily located at the first position (nearest node to intersection zone) on the crossing arm. In practice, the way intralogistics layouts are designed makes it hard to guarantee possiblites for overtaking. Thus we assume that overtaking is not possible in general. This issue gives rise to the problem that F* can be blocked by other AGV with lower order values and a respecting lower position in the scheduling list. To counteract gridlocks based on this problem, the following rule set of a sequential entry system is applied across all approaches, see Table 3.

Table 3: Rule set „Sequential Entry System (SES)“

Rule	Description
(1)	Determine all F with identical R (requestnodeId).
(2)	Identify the blockade group N
(3)	Process this group N as a platoon (anti-blockade protocol)

For the determnination of our platoon for the crossing sequence the following applies:

$$N(F^*) = \{F \in Q \mid R(F) = R(F^*) \wedge T(F) \leq T(F^*)\}$$

i) Priority approach P (earliest due deadline)

In the first extension, vehicles are sequenced under comparison of their priority values. The master control reads the *priorityvalue* variable from the AGV's transit request and inserts it into the crossing queue accordingly. In the case of conflict where competing vehicles have the same priority value further decision parameters are used to allow distinctive decision making. Thus includes $P \in \{1,2,3,4\}$ and $P = 1$ represents the highest priority. For two competing vehicles at different intersection arms the following applies:

$$F_X \prec F_Y \Leftrightarrow \begin{cases} P_X < P_Y & \text{(P1)}, \\ P_X = P_Y \wedge D_X < D_Y & \text{(P2)}, \\ P_X = P_Y \wedge D_X = D_Y \wedge T_X < T_Y & \text{(P3)} \end{cases}$$

P2 allows us to include the orderdeadline as a decision parameter if P1 cannot find a distinctive decision. Altough using more than one parameter the occurrence of a deadlock event is still possible if two competing AGVs have the same *priorityvalue* and the same remaining time to their *orderdeadline*. The control approach thus falls back on the underlying control system with the FCFS principle as a last resort to make a clear decision on right

of way in P3. The sorting mechanism then compares the *timestamp* variable and selects the AGV with the earliest registration on the registration node registration nodespoint (transmitted request).

ii) Queueing approach W (longest queue first)

In this approach, the longest AGV queue should be preferred and given priority. To determine the AGV-queue of an intersection arm, the master control reads the *requestnodeId* variable from the transit requests. With that all AGV with the same Id are grouped (vehicles of the intersection arm that have already been processed are no longer included in the list). For two competing intersection arms the following applies:

$$F_max\ (Q) = (_F\ \in Q\ \hat{}\ argmin)(P, D, T)$$

$$Q_X \prec Q_Y \Leftrightarrow \begin{cases} N_X < N_Y & \text{(W1)}, \\ N_X = N_Y \wedge F_{max}(Q_X) \prec_p F_{max}(Q_Y) & \text{(W2/3)}, \end{cases}$$

Similar to the priority approach this scheduling orientation encounters a conflict, if two competing queues have the same length. For equal lengths, the highest priority vehicle F_{max} within each queue is compared (W2). Thus the priority approach with its three instances is also implemented in the cascadic decision process as a fallback option.
Addtionally when Q_i is selected via W2 / W3 only the part of the queue upon and including F_{max} will get the right of way for transit. Thus counts:

$$N_{transit} = \{ F \in Q_i \mid T(F) \leq T(F_{max}(Q_i)) \}$$

This is primarily because the trailing vehicles do not impede the highest priority vehicle. Moreover, it is probable that, during the course of platoon processing, new queues will form on other arms of the intersection. In the event of the entire intersection arm being processed in accordance with the priorisation function, the fundamental principle of enhancing the overall throughput times for all vehicles would be contravened.

iii) Zone approach Z (constant priorisation)

The third control approach represents a zone-based focus to traffic control. In practice, driving lanes are usually frequented to different degrees. In particular, the main traffic lanes of a intralogistic layout are subject to high vehicle densities. Accordingly, the queueing approach makes sense, but does not guarantee that main streets (H) can generally be given priority over side streets (NB). If a NB queue is the same length as an H queue, the priorities clash and, conversely, there is no guarantee of right of way. If it is not possible to consistently prioritise main streets, there is a risk of congestion and traffic gridlock.

$$Q_X \prec Q_Y \Leftrightarrow \begin{cases} Z_X = Z_H \wedge Z_Y = Z_{NB} & \text{(Z1)}, \\ Z_X = Z_H \wedge Z_Y = Z_H \wedge Q_X \prec_W Q_Y & \text{(Z2)}, \\ Z_X = Z_{NB} \wedge Z_Y = Z_{NB} \wedge Q_X \prec_P Q_Y & \text{(Z3)} \end{cases}$$

This approach also sees a conflict, if more than one arm with the zone assignment Z_H leads into an intersection. In this sense, in Z2 the queueing approach is used for decreasing the average waiting time at the intersection. In light of the higher waiting times that are generally expected on side roads, Z3 makes use of the priority approach directly to minimize the chance for high value orders to miss their deadlines.

iv) Hybrid approach H (shortest processing time)

Lastly, a combined control concept with switching rule sets could be used to prioritise vehicles not only according to order values but also according to their transit speed (not in conflict with system rule 1 because the application only remains until the AGV arrives at the decision node). In this case the priority orientation and a speed orientation (G) are combined. G refers to the Shortest Processing Time (SPT) rule from production scheduling theory. As long as only vehicles with priority values 3 and 4 (where 4 is the lowest of all priorities) are registered at an intersection that represents a bottleneck, the speed orientation counts to maximise the throughput of an intersection in normal operation. If vehicles with priority values 1 and 2 reach the conflict area, the priority orientation takes effect with the next decision run, which then guides the "priority vehicles" through the intersection as quickly as possible.

$$G_{active} \Leftrightarrow \begin{cases} True & if \forall F \in \text{requests}, P(F) \geq 3, \\ False & otherwise \end{cases}$$

As soon as these vehicles have successfully passed the conflict area and the master control only recognises priority values of 3 or 4 in the sorting list again, the system can switch back to the speed-oriented SPT approach with the following decision run.

$$F_X \prec F_Y \Leftrightarrow \begin{cases} S_X < S_Y & if\ G_{active} & \text{(H1)}, \\ F_X \prec_p F_Y & otherwise & \text{(H2)}, \end{cases}$$

3.2.4 Concept of a safety protocol (Deadline fairness)

If new vehicles to be prioritised keep arriving at the intersection arms, there is a risk of starvation for AGVs with a) low priority values, b) in short queues or, c) at zones with side road assignments.
To ensure that those AGVs can still reach their destination by the deadline using the specific control approaches, an external counting factor can be implemented in the decision-making process. This measures the time until the deadline D and sets a threshold for the maximum number of iterations that a request can be in the sort list without being processed. As soon as the threshold is reached, the traffic management system has to take action. When setting the threshold, the planned route of the AGV should also be taken into account. The more intersections a vehicle has to pass on its way, the lower the threshold should be for that vehicle for each conflict area.

Machine learning is a suitable method for determining the respective threshold value. With each executed order, the system learns to better adapt threshold values for the specific layout in the next run. As a matter of logic, as many order scenarios as possible should be simulated before the operational phase so that the definition of the threshold values for the start of regular operation has reached an appropriate level of maturity. The following concept including all operators from Table 4 could be used as a starting point:

Table 4: „List of operators for safety protocol“

Variable	Description
$C(F) \in N$	Count of unresolved decision iterations for vehicle F
$C_{max}(F)$	Deadline-dependent threshold for vehicle F
$C_A(F)$	Intermediate reporting point of the threshold value for vehicle F
$P_{eff}(F)$	Effective priorityvalue (temporary) for vehicle F

The possible application in priority control, speed and queue control is configured by adapting the priorityvalue of an vehicle depending on the unresolved decision iterations of the request in the scheduling list. Thus the simulated priorityvalue is decreased by factor 1 over n number of iterations until the counting factor reaches its maximum. With that the master control has no other option than permitting the request for transit by selecting priorityvalue 1, this measure should be sufficient because until this mark the specific vehicle has the oldest timestamp at the intersection. The intention is not to give direct clearance for transit, as this could lead to the risk of the safety protocol handling more and more vehicles directly over time and thus losing the actual application of the prioritization approaches. However, the efficiency of the protocol stands and falls with the definition of the threshold values.

$$P_{eff}(F) \Leftrightarrow \begin{cases} \mathrm{P(F)} & \text{if } \mathrm{C(F)} < \mathrm{C_A\,(F)} \\ \max\,(1, \mathrm{P(F)} - 1) & \text{if } \mathrm{C_A(F)} \leq \mathrm{C(F)} < C_{max}(\mathrm{F}) \\ 1 & \text{if } \mathrm{C(F)} > C_{max}\,(\mathrm{F}) \end{cases}$$

The possible zone application is configured by adapting the zone assignment of the affected AGV. The assignment should be switched from a side to a main road assignment for resolving the gridlock in the intersection queue, if the amount of unresolved iterations of the request meets or exceeds the threshold.

$$Z_{eff}(F) \Leftrightarrow \begin{cases} Z_H & if\ \mathrm{C(F)} \geq \mathrm{C_A\,(F)} \wedge Z\,(F) = Z_{NB} \\ Z(F) & otherwise \end{cases}$$

After the successful transit the master control can adapt the maximum threshold for the remaining part of the transport order of the AGV and the other vehicles of a possible crossing platoon to counteract the increasing deadline pressure dynamically.

$$C_{max}^{new}(F) \Leftrightarrow \begin{cases} \max(1, C_{max}(\mathrm{F}) - 1\,) & if\ Z(F) = Z_{NB} \\ C_{max}(\mathrm{F}) & otherwise \end{cases}$$

In this case the remaining amount of junctions which the vehicles have to pass on their routes have to be taken into account. As a rule of thumb the threshold can be decreased by factor 1 for training purposes. Furthermore the implementation of an additional priority adaption after the transit is conceivable, but needs to be tested.
Overall the effectiveness of a additional priority based adaption is dependent on the next zone assignment of the affected vehicles, because if they enter a side road zone again, the priority value only matters if side road queues negotiate for the right of way.

4. Initial Validation Case

To get an first overview on the impact of the four new control approaches on their own, a small test scenario is applied. The object under investigation is an intersection with four crossing arms (see Figure 1). Within Table 12 the scenario specific parameters are listed:

Table 4: Parameters for validation

Parameters	Approach 1&2	Approach 3	Approach 4
Arrival rate	[2s, 4s]	[2s, 4s]	[1s, 3s]
Arrival distribution	P1 = 0,05 P2 = 0,10 P3 = 0,70 P4 = 0,15	P1 = 0,05 P2 =0,10 P3 =0,70 P4 =0,15	P1 = 0,05 P2 =0,10 P3 =0,70 P4 =0,15
Simulation time	5 min	5 min	5 min
Crossing time	1m/s	1m/s	[2s,4s]
AGV distribution	Equal	H = 0,5 NB = 0,1667	Equal
Order Deadlines	P1=[25s,50s] P2=[30s,60s] P3=[70s,100s] P4 = D >100	P1=[25s,50s] P2=[30s,60s] P3=[70s,100s] P4 = D >100	P1=[25s,50s] P2=[30s,60s] P3=[70s,100s] P4 = D >100

Each approach is compared individually with the FCFS-principle using four key performance indicators for every AGV priority class: lead time, delay, vehicles on schedule and throughput.
By reason of the different priorisation methods the comparability between the three classes is not given at the moment, rather this could be a future research topic. The results in Appendix 1 show that by implementing priority rule based scheduling decisions specific AGV groups can be temporarily leveraged, but on the other side some vehicles suffer in their performance. In total approach 4 achieved the best results in view of the majority of AGV in the scenario. The successful attempt to combine two scheduling approaches with a different focus in parallel and thereby unite their strengths promises a starting position that can be built on further. The hybrid approach thus represents a functional dynamic control concept that stands out among the control approaches in this research paper. In detail approach 4 increases the total throughput from 85 percent of AGV by about 2 percent, additionally the lead time, average delay and the percentage of on schedule vehicles are enhanced by 10 to 15 percent.

5. Conclusion and Outlook

To counteract the lack of responsiveness of the current control principle in the future, the implementation of coordination extensions based on rule-based decisions is suitable as the first validation has shown. The ability to extend the static character of the FCFS principle with priority-rules based on scheduling methods from the production field makes it possible to achieve the desired flexibility in the event of dynamic traffic situations and in regards to handling high volumes of vehicles. It is important to note that conflict areas differ in their structure and functionality. As a result, control approaches also differ in their coordination effectiveness for specific AGV-groups. Therefore, it is recommended not to rely exclusively on one control approach, as shown, for controlling the fleet management system, but rather to use hybrid coordination methods if challenges arise in the traffic flow. In addition, it is conceivable to consult further sequencing rules and to create new control methods from their combination. Based on the constructed templates of this paper, it is also possible to develop an individual configuration for specific layouts. This capability could help companies to take matters into their own hands after a successful first implementation. Finally, with regard to the feasibility of this recommendation, it should be noted that the control methods are designed in such a way that they can be embedded in the background of current centrally oriented control systems. Their activation can be integrated for as long as its necessary or useful. This also helps to limit the increasing complexity for the central control system until decentralized methods achieve the desired level of maturity and spread in industrial applications.
With further regard there are still topics for additional research.First of all in this paper the extension for a systemwide route analysis mechanism couldn't be realized by now. Its implementation could deliver a key tool to gain deeper insights into route and scheduling predicition, which are essential for a successful implementation of the safety protocol and ongoing coordination efficency. Furthermore after the consideration of our simulation results, the question of the "most efficient" coordination approach for intersections stays open. It therefore remains to be critically questioned to what extent the results from the limited test scenarios can also be transferred to other traffic situations. In order to achieve more accurate statements on the efficiency of the presented coordination approaches field testing in simulation environments is needed. Only after successful validation the consideration should be given to implementing the control extension. Within the tests, further questions and open points need to be answered that could not be identified within the scope of this work:

- The various asset categories and their structure cannot be uniformly defined, so the conflict area structure developed must be transferred to other conflict areas.
- For the transition of the control methods, clear trigger and stop conditions must be formulated (this point could be particularly complicated if more than two control methods are used).
- Although the pilot of the Deadline Fairness Protocol could be configured, the specific threshold values for the individual priority values still need to be determined.
- The observation horizon currently only takes into account the first five queue positions of an EntryPoint in the decision run. This means that if the queue under consideration has five AGVs with priority value 4 waiting at an Entry Point, the DFP effects late. As long as no position becomes free, for example, the next AGV in line (usually queue position 6) with priority value 1 cannot be recorded in the decision run and therefore the DSP cannot intervene. The bottleneck vehicle therefore has no chance of meeting its deadline. A method must be found to counteract this problem in the future.

The results also raise the question of whether it is less the underlying control strategy and more the maxAgvCount of 1 that leads to the restricted performance (throughput) of the intersection. As long as only one AGV can pass the intersection and, conversely, only one intersection arm/access route is served in parallel, an improvement in specific performance factors can only be achieved to a limited extent, even with dynamic control approaches. This statement is supported by the test runs of the initial approaches, as no increase in throughput could be achieved for the overall system even with prioritization procedures. At last the consideration of the trajectory and the envelope curve could play a decisive role in possibly serving more than one crossing arm at the same time. This is because, as long as AGVs from different access routes do not collide during their transit, several AGVs could pass through the intersection at the same time in the future. Consequently, the efficiency of the coordination could increase and the application of the envelope curve could show a further improvement to the dynamic control concepts in this work. However, the implementation of this new concept would require further adaptation of the conflict area into smaller zones and the establishment of a timed sequence plan. The resulting complexity for a central control system could be too high with the use of envelopes and trajectories, especially in large layouts, so this proposal must be compared with a completely decentralized method.

References

Aguiar, G.T.; Oliviera, G.A.; Tan, K.H.; Kazantsev, N. and D. Setti. 2019. "*Sustainable implementation success factors of AGVs in the Brazilian industry supply chain management*." Procedia Manufacturing 39, Elsevier, Amsterdam, Netherlands, pp.1577-1586.

Bashiri, M. and C. Flemming. 2017. „*A Platoon-Based Intersection Management System for Autonomous Vehicles*." Proceedings of the IEEE Intelligent Vehicles Symposium (IVS), Band 4. IEEE, New York, United States, pp. 667–672.

Basile, F.; P. Chiacchio; and E. Di Marino. 2019. „*An Auction-Based Approach to Control Automated Warehouses Using Smart Vehicles*.“ Control Engineering Practice 90. International Federation of Automatic Control, Laxenburg, Austria, pp. 285–300.

Carlino, D.; S.D. Boyles; and P. Stone. 2013. „*Auction-Based Autonomous Intersection Management*.“ Proceedings of the International IEEE Conference on Intelligent Transportation Systems (ITSC), Band 16. IEEE, The Hague, Netherlands, pp. 529–534.

Coffman, E.; M. Elphick; and A. Shoshani. 1971. „*System Deadlocks*.“ ACM Computing Surveys (CSUR) 3, Nr. 2 (Jun). Association for Computing Machinery, New York, United States, pp. 67–78.

Connette, C. 2013. *Kinematische Modellierung und Regelung omnidirektionaler, nicht- holonomer Fahrwerke*. In: Bauernhansl, Thomas; Verl, Alexander und Westkämper, Engelbert (Hrsg.): Stuttgarter Beiträge zur Produktionsforschung. Volume 12, University Stuttgart, Fraunhofer IPA, Stuttgart, Germany

De Ryck, M.; M. Versteyhe; and F. Debrouwere. 2020. „*Automated Guided Vehicle Systems, State-of-the-Art Control Algorithms and Techniques*.“ Journal of Manufacturing Systems 54. Elsevier, Amsterdam, Netherlands, pp. 152–173.

Deloitte . 2023. *Digital Readiness Report 2023*.

Deutscher Bundestag. 2020. „*Ursachen von Verkehrsstaus*.“ WD 5: Economy and Transport, Nutrition, Agriculture, and Consumer Protection.

Dharmasiri, P.; I. Kavalchuk; and M. Akbari. 2019. „*Implementation Traffic Control Algorithm for Multi-AGV System*.“ Proceedings of the 9th International Conference on Operations and Supply Chain Management (OSCM). Ho Chi Minh City, Vietnam, pp. 1–7.

Dickmann, P. 2015. „*Lean Material Flow: with Lean Production, Kanban, and Innovations*. “ 3rd ed. Springer Vieweg, Berlin, Heidelberg, Germany.

Dilefeld, M. 2023. „*Challenges in the Design of Mobile Robots (AGVs and AMRs)*.“ Proceedings of the ASIM Symposium on Simulation in Production and Logistics 2023, edited by S. Bergmann, N. Feldkamp, R. Souren, und S. Staßburger. University Press Ilmenau, Germany.

Dresner, K. and Stone, P. 2006. “*Traffic Intersections of the Future*”. 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference, Boston, United States, pp.1593-1596.

Dresner, K. and P. Stone. 2008. „*A Multiagent Approach to Autonomous Intersection Management*.“ Journal of Artificial Intelligence Research (JAIR) 31. Association for the Advancement of Artificial Intelligence, Washington, D.C., United States, pp. 591–656.

Fottner, J.; D. Clauer; and F. Hormes et al. 2021. „*Autonomous Systems in Intralogistics – State of the Art and Future Research Challenges*.“ Logistics Research 14, Nr. 1. BVL, Brunswick, Germany.

Fragapane, G.; R. de Koster; and F. Sgarbossa et al. 2021. „*Planning and Control of Autonomous Mobile Robots for Intralogistics: Literature Review and Research Agenda*.“ European Journal of Operational Research (EJOR) 294, Nr. 2. Elsevier, Amsterdam, Netherlands, pp. 405–426.

Guney, M.A. and I.A. Raptis. 2021. „*Dynamic Prioritized Motion Coordination of Multi-AGV Systems*.“ Robotics and Autonomous Systems 139. Elsevier, Amsterdam, Netherlands.

Günthner, W.A.; M. ten Hompel; und P. Tenerowicz-Wirth et al. 2012. „*Algorithms and Communication Systems for Cellular Conveying Technology.“* Research Report IGF Project 16166 N of the German Logistics Association (BVL) e.V. Chair for Material Handling, Material Flow, Munich, Dortmund, Germany.

Hertzberg, J.; K. Lingemann; und A. Nüchter. 2012. *Mobile Robots: An Introduction from a Computer Science Perspective*. Springer, Berlin, Heidelberg, Germany.

Kagermann, H. 2014. „*Chancen von Industrie 4.0 nutzen*.“ In: Bauernhansl, T., ten Hom- pel, M., Vogel-Heuser, B (Hsg.) Industrie 4.0 in Produktion, Automatisierung und Logistik, Springer Vieweg, Wiesbaden, Germany, p.603-614.

Khayatian, M.; M. Mehrabian; and E. Andert et al. 2020. „*A Survey on Intersection Management of Connected Autonomous Vehicles*.“ ACM Transactions on Cyber-Physical Systems 4. Association for Computing Machinery, New York, United States

Kubasakova, I.; J. Kubanova; and D. Benco et al. 2024. „*Implementation of Automated Guided Vehicles for the Automation of Selected Processes and Elimination of Collisions between Handling Equipment and Humans in the Warehouse*.“ Sensors 24, Nr. 3, Article 1029. MDPI, Basel, Switzerland.

Le-Anh, T. and R. de Koster. 2006. „*A Review of Design and Control of Automated Guided Vehicle Systems*.“ European Journal of Operational Research (EJOR) 171, Nr. 1. Elsevier, Amsterdam, Netherlands, pp. 1–23.

Lu, W.; S. Guo; and S. Tao et al. 2021. „*Analysis of Multi-AGVs Management System and Key Issues: A Review*.“ Computer Modeling in Engineering & Sciences 131, Nr. 3. Tech Science Press, Henderson, United States, p. 1197–1227.

Niels, T. 2022. „*Integrated Intersection Control for Connected Automated Vehicles, Pedestrians, and Bicyclists*.“ Technical University of Munich, Germany.

Nikelowski, L. and Wolny, Michael. 2020. „*Der Weg zur Smart Factory.“* Future Challenges in Logistics and Supply Chain Management, Volume 15, Institut für Materialfluss und Logistik, Fraunhofer IML, Dortmund, Germany, p.1-15.

Pichler, T. 2014. „*Comparison of Automated Guided Vehicle Systems in Production, Intralogistics, and Service Logistics*.“ Montan University Leoben, Leoben, Austria.

Pratissoli, F.; R. Brugioni; N. Battilani; und L. Sabattini. 2023. „*Hierarchical Traffic Management of Multi-AGV Systems with Deadlock Prevention Applied to Industrial Environments*.“ IEEE Transactions on Automation Science and Engineering. IEEE, New York, United States.

Schmidt, T.; K.-B. Reith; N. Klein; et al. 2020. „*Research on Decentralized Control Strategies for Automated Vehicle-based In-house Transport Systems – a Survey*.“ Logistics Research 13. BVL, Brunswick, Germany, p. 1–13.

Scholz, M. 2019. „*Intralogistics Execution System with Integrated Autonomous, Service-Based Transport Entities*.” FAU Studies in Mechanical Engineering, Vol. 331. FAU University Press, Erlangen, Nuremberg, Germany.

Schwarz, C.; J. Schachmanow; J. Sauer; et al. 2013. „*Self-Controlled Automated Guided Vehicle Systems*.“ Logistics Journal 12. BVL, Brunswick, Germany, p. 1–10.

Siegfried, P. and R. Bourafa. 2023. „*A Review of the Automated Guided Vehicle Systems: Dispatching Systems and Navigation Concept*." Automobile Transport 52. Kharkiv National Automobile and Highway University, Kharkiv, Ukraine, p. 80–88.

Ullrich, G. and T. Albrecht. 2023. „*Automated Guided Vehicle Systems: The AGV Guide – Technology, Practical Applications, Planning, and History"*. 4th ed. Springer Vieweg, Wiesbaden, Germany.

VDA. 2022. „*Interface for Communication between Automated Guided Vehicles (AGVs) and a Central Control System*." Version 2.0.0.

Voß, T. 2023. „*Method for Dynamic Adaptation of Sequence Rules using Reinforcement Learning"*. University of Lüneburg, Germany.

Zheng, Z., Ahn, S., Chen, D., and Laval, J. 2021. "Applications of wavelet transform for analysis of freeway traffic: Bottlenecks, transient traffic, and traffic oscillations." Transportation Research Part B: Methodological 113. Elsevier, Amsterdam, Netherlands.

Appendix 1

								FCFS		Approach Z + SES						
		FCFS	Approach P + SES	Delta	FCFS	Approach W + SES	Delta	H	NB	H	NB	Delta (H)	Delta (NB)	FCFS	Approach G + SES	Delta
Lead Time (in sec)		43,00	38,18	-12,62	43,00	41,73	-2,95	41,48	44,68	27,43	53,16	-51,22	15,95	56,00	20,80	-62,50
	Priority 1	45,40	28,51	-59,24	45,40	41,71	-8,13	47,67	49,89	25,67	61,90	-85,70	-85,70	45,00	12,00	-77,78
	Priority 2	43,50	34,66	-25,50	43,50	41,03	-5,68	26,61	50,64	18,17	58,71	-46,45	-46,45	54,00	12,73	-75,93
	Priority 3	42,20	44,17	4,46	42,20	42,64	1,04	45,42	44,42	35,84	58,22	-26,73	-26,73	57,00	23,56	-57,89
	Priority 4	41,10	45,37	9,41	41,10	41,56	1,12	46,23	33,75	30,03	33,79	-53,95	-53,95	69,00	34,90	-49,28
Delay (in sec)		39,00	33,93	-13,00	39,00	37,65	-3,46	37,45	40,67	23,45	49,07	-59,70	-59,70	58,00	17,79	-69,33
	Priority 1	41,40	24,51	-40,80	41,40	37,71	-8,91	43,67	45,89	21,67	57,89	-101,52	-101,52	42,00	9,00	-78,57
	Priority 2	39,50	30,36	-23,14	39,50	36,69	-7,11	22,61	46,64	14,17	54,74	-59,56	-59,56	51,00	10,06	-80,27
	Priority 3	38,50	40,17	4,34	38,50	38,62	0,31	41,32	40,39	31,93	53,90	-29,41	-29,41	73,00	20,11	-72,45
	Priority 4	36,70	40,69	10,87	36,70	36,56	-0,38	42,22	29,75	26,04	29,77	-62,14	-62,14	66,00	32,00	-51,52
On Schedule (in percent)		69,50	77,39	11,35	69,50	74,06	6,56	76,37	54,50	77,40	51,16	1,33	1,33	59,00	99,00	67,80
	Priority 1	50,30	73,89	46,90	50,30	59,17	17,63	50,00	33,33	50,00	11,10	0,00	0,00	25,00	100,00	300,00
	Priority 2	49,80	61,45	23,39	49,80	59,84	20,16	80,56	48,89	86,11	39,44	6,45	6,45	54,50	100,00	83,49
	Priority 3	77,80	74,21	-4,61	77,80	77,23	-0,73	74,93	68,90	91,40	54,10	18,02	18,02	54,10	98,60	82,26
	Priority 4	100,00	100,00	0,00	100,00	100,00	0,00	100,00	100,00	100,00	100,00	0,00	0,00	100,00	100,00	0,00
Throughput (units/AGV)		75,00	75,00	0,00	75,00	75,00	0,00	39,00	36,00	40,33	34,67	3,30	3,30	100,00	103,00	3,00
	Priority 1	6,00	6,67	11,17	6,00	5,67	-5,50	1,30	2,30	1,30	2,30	0,00	0,00	4,00	5,00	25,00
	Priority 2	8,70	9,33	7,24	8,70	8,67	-0,34	3,70	5,00	3,70	4,00	0,00	0,00	11,00	14,00	27,27
	Priority 3	50,70	50,33	-0,73	50,70	51,00	0,59	28,00	25,70	29,00	25,00	3,45	3,45	74,00	74,00	0,00
	Priority 4	9,70	8,67	-10,62	9,70	9,67	-0,31	6,00	3,00	6,30	2,70	4,76	4,76	12,00	11,00	-8,33

Natural Language Processing (NLP) mit PyTorch am Beispiel eines Hochschul-Chatbots

Nicolas Lang (M.Sc.)

Technische Hochschule Mittelhessen
Fachbereich Mathematik, Naturwissenschaften und Datenverarbeitung (MND)
Wilhelm-Leuschner-Str. 13, 61169 Friedberg
nicolas.lang@mnd.thm.de

Prof. Dr. Harald Ritz

Technische Hochschule Mittelhessen
Fachbereich Mathematik, Naturwissenschaften und Informatik (MNI)
Wiesenstr. 14, 35390 Gießen
harald.ritz@mni.thm.de

Prof. Dr. Frank Kammer

Technische Hochschule Mittelhessen
Fachbereich Mathematik, Naturwissenschaften und Informatik (MNI)
Wiesenstr. 14, 35390 Gießen
frank.kammer@mni.thm.de

Jonas Wölfer (M.Sc.)

Technische Hochschule Mittelhessen
Fachbereich Mathematik, Naturwissenschaften und Informatik (MNI)
Wiesenstr. 14, 35390 Gießen
jonas.woelfer@mni.thm.de

ABSTRACT

Der vorliegende Artikel beschreibt die Weiterentwicklung des KI-gestützten FAQ-Chatbots der Technischen Hochschule Mittelhessen (THM) zur Verbesserung der Antwortgenauigkeit durch die Entwicklung eines domänenspezifischen Transformers für das Extractive Question Answering (EQA).

Im Jahr 2021 wurde der Chatbot „Winfy“ im Rahmen eines Masterprojekts ins Leben gerufen. Dieses Projekt wurde vom Prüfungsausschussvorsitzenden B.Sc. und M.Sc. Wirtschaftsinformatik angestoßen, um den Studierenden eine Möglichkeit zu bieten, Antworten auf häufig gestellte Fragen (FAQ) rund um das Thema Prüfungsangelegenheiten in den beiden Wirtschaftsinformatik-Studiengängen zu erhalten.

Der Chatbot basiert auf einem Fragen-Antwort-Katalog, in dem eine einzelne Antwort mehreren Fragen zugeordnet ist. Wenn ein Studierender eine Frage stellt, liefert das System den Antwortblock, der mit der semantisch am besten passenden Frage verknüpft ist. Dies kann jedoch die Suche nach der exakt gewünschten Information erschweren. Um das Antwortverhalten zu verbessern, wurde ein Extractive Question Answering (EQA)-Modell entwickelt. Dieses sollte gezielt die Antwort aus dem Antwortblock extrahieren.

Hierfür wurde ein Transformer-Encoder mit ELECTRA-Architektur entwickelt und in einem zweistufigen Verfahren trainiert: Zunächst erfolgte ein Pre-Training auf domänenspezifischen Texten, gefolgt von einer Feinabstimmung auf einem speziell erstellten und annotierten EQA-Datensatz für den Hochschulbereich. Abschließend wurden verschiedene deutsche Sprachmodelle auf diesem und anderen etablierten EQA-Datensätzen verglichen. Das Modell „Winfy-ELECTRA-Base“ erzielte dabei die besten Ergebnisse und stellt eine signifikante Verbesserung der Antwortqualität dar.

SCHLÜSSELWÖRTER

Chatbot, Machine Learning (ML), Natural Language Processing (NLP), Transformers, Huggingface, Neuronale Netze, Frequently Asked Questions (FAQ), PyTorch, ELECTRA, Extractive Question Answering (EQA), Künstliche Intelligenz (KI)

VERWANDTE ARBEITEN

Verwandte Arbeiten verfolgen ebenfalls den Ansatz, domänenspezifisches Pre-Training einzusetzen, um die Leistungsfähigkeit der Modelle zu steigern, und zeigen dabei, dass dies oft bessere Ergebnisse erzielt als die Feinabstimmung allgemein vortrainierter Modelle. Zu bekannten Beispielen zählen „SciBERT“ oder „PubMedBERT“, die auf wissenschaftlichen Texten trainiert wurden, mit einem kontextspezifischen Vokabular arbeiten und somit deutlich besser abschneiden [1] [2].

AUSGANGSSITUATION

Um die Mitarbeitenden des Fachbereichssekretariats MND der Technischen Hochschule Mittelhessen (THM) zu entlasten, wurde 2021 das Projekt „Winfy“ vom Prüfungsausschussvorsitzenden gestartet. Dieser KI-basierte FAQ-Chatbot (https://feedback.mni.thm.de/winfy/) wurde entwickelt, um häufig gestellte Fragen zu Prüfungsangelegenheiten im Studiengang Wirtschaftsinformatik zu beantworten. Der Chatbot arbeitet mit einer Datenbank häufig gestellter Fragen, wobei mehrere

Fragen mit einer Antwort verknüpft sind. Durch eine Ähnlichkeitsberechnung, die vom Sentence-Transformer „German-Semantic“ [3] durchgeführt wird, wird die eingehende Nutzerfrage mit dem vorhandenen Fragenkatalog abgeglichen. Die Antwort auf die Frage mit dem höchsten Ähnlichkeitswert wird ausgegeben [4].

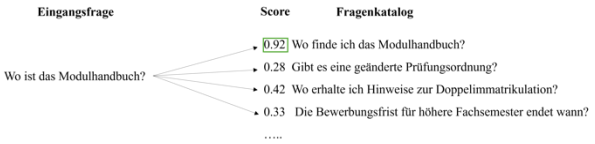


Abbildung 1: Ähnlichkeitsberechnung Sentence-Transformer

Das „German-Semantic“-Modell erzielt bereits sehr gute Ergebnisse. In den Antwortblöcken sind die Antworten auf mehrere Fragen zusammengefasst. Dies spart Zeit bei der Pflege der Fragen und kann den Studierenden zusätzliche Informationen im selben Kontext geben, erschwert allerdings die Antwortfindung, da die Länge der Antworten sehr umfangreich sein kann.

AUFGABENSTELLUNG UND ZIELSETZUNG

Das Ziel war es, die Antwortgenauigkeit des Chatbots zu verbessern. Dafür sollte ein eigenes neuronales Netz auf Basis der Transformer-Architektur für den Hochschuleinsatz gebaut werden. Als Einschränkung galt, dass das System keinen Text generieren durfte, um zu vermeiden, dass Falschinformationen verbreitet werden. Somit wurde der Ansatz des Extractive Question Answering verfolgt [17]. Anstelle eines bereits vorhandenen Modells sollte ein eigenes trainiert werden, um dieses auf den domänenspezifischen Kontext anzupassen. Dafür mussten entsprechende Daten für das Pre-Training gesammelt, gefiltert und aufbereitet werden.

Da kein EQA-Datensatz für die Hochschuldomäne existiert, um abschließend die Performance des Modells zu bewerten, musste selbst einer erstellt und annotiert werden. Als Basis wurde der bestehende Fragen-Antwort-Katalog des Chatbots genutzt.

Abschließend sollten die bereits vorhandenen Modelle mit dem selbst trainierten Modell verglichen werden – zum einen anhand anerkannter EQA-Benchmarks, aber auch auf dem domänenspezifischen Datensatz.

DATENSAMMLUNG

Für das Pre-Training musste eine große Menge an Daten gesammelt werden. Es wurde auf bekannte Quellen wie Wikipedia [5] und Wikivoyage [6] zurückgegriffen. Diese enthalten hochwertige Texte mit langen Sequenzen und eignen sich sehr gut für das Training von neuronalen Netzen.

Um an domänenspezifische Hochschultexte zu gelangen, wurden Modulhandbücher und Prüfungsordnungen gesammelt und in einem Datensatz namens „Academic Crawl“ zusammengefasst.

Da dies nur zu einer sehr geringen Datenmenge führte, wurde außerdem der GC4-Datensatz [7] verwendet. Dabei handelt es sich um den deutschen Auszug des CommonCrawl [8], der Texte von Internetseiten, Nachrichtenseiten sowie Foren und sozialen Medien enthält. Der GC4-Datensatz ist in drei Qualitätskategorien unterteilt: Head, Middle und Tail [7]. Es wurden nur die Artikel der Head-Kategorie verwendet, um den Qualitätsgrad hochzuhalten. Weiterhin wurden anschließend lediglich die Artikel verarbeitet, die eine Qualität von 0,99 oder höher aufwiesen. Dies ergab eine Datenmenge von 450 GB.

Da auch domänenfremde Texte enthalten waren, mussten diese gefiltert werden. Dies wurde aufgrund der enormen Datenmenge mithilfe einer Methode des Information Retrieval, dem TF-IDF (Term Frequency-Inverse Document Frequency), durchgeführt [9]. Dabei wurden exemplarische Hochschultexte genommen und mit den Artikeln des Datensatzes verglichen. Als Maßgabe sollten die Artikel eine Ähnlichkeit von 40% zu den Beispielen aufweisen. Nach der Filterung hatte der Datensatz eine Größe von 16 GB. Schlussendlich wurde das Vorab-Training mit folgenden Datensätzen durchgeführt:

Datensatz	Größe
Wikipedia 2024	6,2 GB
Wikivoyage	55 MB
Crawled Academic Dataset	6,1 MB
GC4 Corpus Filtered	16 GB
Total	**22,25 GB**

Tabelle 1: Gesammelte Pre-Training-Datensätze

Um die insgesamt 22,25 GB an Daten für das Pre-Training zu nutzen, mussten diese noch aufbereitet werden.

DATENAUFBEREITUNG

Die Daten wurden in drei Schritten aufbereitet: der Satztrennung, Normalisierung und Prüfung auf unerlaubte Zeichen. Transformer-Encoder arbeiten mit einer fixen Eingabelänge. Dieses Modell sollte mit einer Tokenlänge von 512 Tokens trainiert werden. Dafür war es wichtig, dass die Trainingsbeispiele diese Länge nicht überschreiten. Um dies zu ermöglichen, mussten die Eingabesequenzen der Datensätze zunächst in einzelne Sätze unterteilt werden, damit diese später auf die gewünschte Länge konkateniert werden können.

Die Satztrennung wurde mithilfe des „SoMaJo“-Tokenizers durchgeführt. Dieser eignet sich besonders gut für die Verarbeitung deutscher Webtexte [10].

Um das Rauschen in den Daten zu verringern und das Vokabular des Tokenizers so effizient wie möglich zu gestalten, wurden die Sätze normalisiert. Dafür wurden die Skripte des CCNet-Projekts [11] verwendet. Dabei wurden Zeichen, die die gleiche Bedeutung haben, auf eine einheitliche Form gebracht, z. B. verschiedene

Formen von Anführungszeichen («, “, '). Weiterhin wurden Unicode-Fehler korrigiert und Emojis entfernt, die häufig in Webtexten auftreten. Somit konnte die Datenqualität angehoben werden.

Weiterhin wurden Zeichen entfernt, die im späteren Einsatz des Sprachmodells nicht vorgesehen waren und somit nicht zum Lernerfolg des Modells beitrugen. Vorgesehene Zeichen waren ASCII-Zeichen sowie ausgewählte Symbole und Umlaute der deutschen Sprache. Durch das Normalisieren und das Entfernen unerlaubter Zeichen sollte sichergestellt werden, dass keine Vokabularplätze des Tokenizers für Tokens verschwendet werden, die nur selten in den Datensätzen vorkommen bzw. im späteren Einsatz überhaupt nicht mehr benötigt werden.

Anschließend wurde der Tokenizer mithilfe der vorliegenden Texte trainiert, damit dieser optimal auf die Trainingsdaten abgestimmt ist.

```
1.  "vocab": {
2.    "[UNK]": 0,
3.    "[PAD]": 1,
4.    "[CLS]": 2,
5.    "[SEP]": 3,
6.    "[MASK]": 4,
7.    "!": 5,
8.    "\"": 6,
9.    "'": 7,
10.   "(": 8,
11.   ")": 9,
12.   ",": 10,
13.   "-": 11,
14.   ".": 12,
15.   "0": 13,
16.   "1": 14,
17.   "2": 15,
18.   ...
19.   "fachober": 29996,
20.   "tschern": 29997,
21.   "##dekan": 29998,
22.   "schleich": 29999
23. }
```

Listing 1: Tokenizer Vokabularauszug

Somit konnten die einzelnen Sätze zur gewünschten Tokenlänge von 512 Tokens zusammengefügt werden. Dabei wurden nur Texte zusammengefügt, die aus dem gleichen Artikel stammten.

EQA-DATENSÄTZE

Um den Modellen das EQA beibringen zu können und deren Performance darauf zu bewerten, mussten Datensätze gesammelt werden. Dabei wurde die maschinelle Übersetzung des SQuAD (Stanford Question Answering Dataset) (hier als SQuAD-De aufgeführt) verwendet. Dieser wurde als Anhang im MLQA-Datensatz [12] von Facebook veröffentlicht. Der Vorteil ist, dass dieser mit seinen 89.996 Einträgen sehr umfangreich ist. Durch die maschinelle Übersetzung leidet allerdings die Datenqualität, da diese nicht die Nuancen der deutschen Sprache enthält [13].

Als weiterer Datensatz wurde der GermanQuAD [13] (German Question Answering Dataset) verwendet. Dieser beinhaltet deutlich weniger Einträge (13.722), weist allerdings eine sehr hohe Qualität auf. Bei den annotierten Texten handelt es sich um originale deutsche Artikel, die von geschultem Personal für diese Aufgabe nach definierten Qualitätskriterien gelabelt wurden [13].

Um die Performance der Modelle für das EQA auf der Hochschuldomäne bewerten zu können, musste noch ein Datensatz erstellt werden, da ein solcher bis zu diesem Zeitpunkt nicht existierte. Als Datenbasis wurde der Fragenkatalog des Chatbots „Winfy“ genutzt.

Dabei wurden die Fragen zuerst bereinigt. Fragen, die sich zu ähnlich waren, wurden automatisiert entfernt (semantische Ähnlichkeit von 85%). Die verbliebenen 2.863 Fragen wurden mithilfe der Qualitätskriterien des GermanQuAD annotiert. Für das Annotieren wurde die Software Label Studio [14] verwendet.

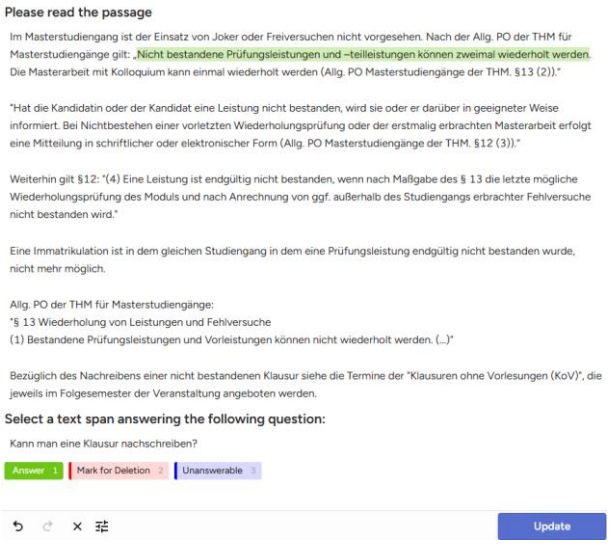


Abbildung 2: Label Studio Annotationsprozess

Dabei wurde die Antwort zur gestellten Frage im Kontext markiert. Fragen, die nicht den Qualitätskriterien entsprachen oder sich nicht beantworten ließen, wurden entsprechend markiert und entfernt.

```
1.  {
2.   "paragraphs": [
3.       {
4.         "context": "Bei Abholung der Chipkarte im InfoCenter wird Ihnen...",
5.         "document_id": 33135,
6.         "qas": [
7.          {
8.           "question": "Wo finde ich die Aktualisierungstermials …"
9.           "id": 33135,
10.          "answers": [
11.                    {
12.            "answer_id": 11195,
13.            "text": "im InfoCenter und in den Gebäuden Gebäuden A2, A4 und A7",
14.            "answer_start": 248
15.            }
16.            ],
17.          "is_impossible": false
18.      }
19.     ]
20.   }
21.  ]
22. }
```

Listing 2: WinfyQuAD

Anschließend wurden die Daten mithilfe eines Skripts in eine für das EQA trainierbare Form gebracht (siehe Abbildung 2 oben). Es verblieben 1.746 Fragen. Der entstandene Datensatz wurde aufgrund der Datenbasis „WinfyQuAD“ (Winfy Question Answering Dataset) genannt.

MODELLARCHITEKTUR

Als Modellarchitektur wurde der ELECTRA-Ansatz (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) gewählt. Dabei werden zwei Modelle gleichzeitig trainiert: ein Generator- und ein Diskriminator-Modell [15].

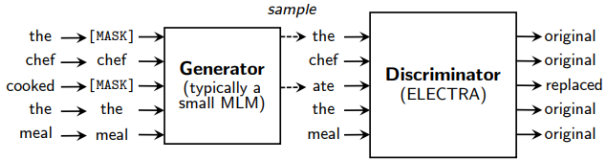


Abbildung 3: ELECTRA-Training (Quelle: [15])

Die beiden Modelle werden mithilfe von Self-Supervised-Learning trainiert. Zuerst werden 15% der Tokens einer Eingabesequenz maskiert. Die maskierte Sequenz wird an den Generator übergeben. Dieser hat die Aufgabe des Masked Language Modeling (MLM) und muss erraten, welche Tokens ursprünglich maskiert wurden. Die Vorhersage des Generators durchläuft einen „Sample"-Prozess, sodass nicht automatisch die wahrscheinlichste Vorhersage des Generators genommen wird. Der Diskriminator erhält die „beschädigte" Eingabesequenz und muss erraten, welche der Tokens echt oder vertauscht wurden. Dadurch muss der Diskriminator bei 100 % der Tokens eine Vorhersage treffen, während dies beim MLM nur bei 15 % geschieht. Dies führt dazu, dass ELECTRA ein dateneffizienterer Trainingsansatz ist als z. B. BERT [15].

Weiterhin besitzt der Generator nur ca. ein Viertel bis ein Drittel der Parametergröße des Diskriminators, damit dieser nicht zu gute Vorhersagen beim MLM ausgibt und das Training aus dem Gleichgewicht gerät [15]. Als Modellgröße wurden folgende Hyperparameter gewählt:

Hyperparameter	Generator	Discriminator
Hidden Layers	12	12
Sequence length	512	512
Hidden Size	256	768
FFN inner hidden size	1024	3072
Attention heads	4	12
Attention head size	64	64
Embedding size	768	768
Learning rate decay	Linear	Linear
Warmup steps	10K	10K
Learning rate	2e-4	2e-4
Attention dropout	0.1	0.1
Dropout	0.1	0.1
Weight decay	0.01	0.01
Batch size	256	256
Train steps	766K	766K
Vocab size	30K	30K
Total parameter	33M	110M

Tabelle 2: Hyperparameter Winfy-ELECTRA-Base

Dabei wurde sich am originalen ELECTRA orientiert.

Um die Architektur umzusetzen, wurde PyTorch in Kombination mit der Transformers-Bibliothek von Huggingface [16] verwendet. Diese Bibliothek stellt bereits die Architektur für den Generator und den Diskriminator bereit. Die Interaktion zwischen diesen beiden Modellen musste allerdings selbst entwickelt werden.

PRE-TRAINING

Das Pre-Training wurde auf einer NVIDIA GeForce RTX 4090 durchgeführt. Die 766.000 Trainingsschritte dauerten insgesamt 560 Stunden.

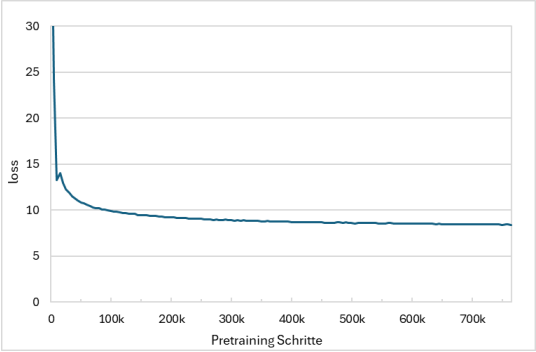


Abbildung 4: Verlustkurve Winfy-ELECTRA-Base

In der obigen Grafik ist der vereinte Validationsverlust der Modelle zu sehen. Dabei wurde der Verlust des Diskriminators mit dem Faktor 50 gewichtet, damit dieser bei der Anpassung der Gewichte bevorzugt wird. Es zeigte sich, dass der Verlust bis zum Ende stetig sank und kein Anzeichen einer Überanpassung auf die Trainingsdaten (Overfitting) zeigte. Somit könnte das Pre-Training auch weiter fortgeführt werden.

Das resultierende Modell wurde nach dem Chatbot benannt, in welchem es verwendet werden soll: „Winfy-ELECTRA-Base". Nun musste dem Modell noch die Aufgabe des EQA angelernt werden.

EXTRACTIVE QUESTION ANSWERING

EQA gehört zu den wichtigsten Aufgaben des Natural Language Processing (NLP). Dabei wird zwischen „Single-Span" und „Multi-Span" differenziert. Beim „Single-Span" wird versucht, die Antwort aus genau einer Textspanne zu extrahieren. Beim „Multi-Span" wird die Antwort aus mehreren Stellen im Text extrahiert.[17]. In diesem Projekt wurde aufgrund mangelnder Verfügbarkeit entsprechender Datensätze nur das „Single-Span"-EQA behandelt.

Um einem Modell das EQA beizubringen, muss die Architektur nicht geändert werden. Es muss lediglich die Ausgangsschicht für den neuen Einsatzbereich angepasst werden. Für die Aufgabe des EQA werden zwei lineare Schichten benötigt. Die erste Schicht soll den Anfang der Antwort im Kontext vorhersagen, die zweite Schicht soll das Ende der Antwort bestimmen [18].

Die Eingabeschicht muss nicht geändert werden, da der Tokenizer damit umgehen kann. Dabei wird das Klassifikationstoken (CLS) verwendet, um den Beginn der Eingabesequenz zu kennzeichnen, und das Trenn-Token (SEP), um die Frage vom Kontext zu trennen

sowie das Ende der Sequenz zu markieren. Das Modell muss nun eine Klassifikationsaufgabe lösen, um die korrekte Antwortspanne vorherzusagen [18].

Um die Laufzeiteffizienz der Modelle bewerten zu können, wird auf zwei Metriken zurückgegriffen: den Exact Match (EM) und den F1-Wert. EM misst, ob das Modell exakt die richtige Antwortspanne vorhersagt. F1 gibt an, wie viele Prozent der vorhergesagten Tokens mit der tatsächlich korrekten Antwortspanne übereinstimmen [19].

FEINABSTIMMUNG

Für den Vergleich wurden vorhandene deutsche Transformer-Encoder Modelle gesucht, um diese mit dem Winfy-ELECTRA-Base zu vergleichen. Dabei standen die beiden Modelle deepset/gbert-base und deepset/gelectra-base zur Verfügung [20].

Für die Feinabstimmung wurden die Modelle, wie auch im Artikel „Improving Non-English Question Answering and Passage Retrieval“ [13] beschrieben, zuerst auf dem SQuAD-De „aufgewärmt“, da dies die Ergebnisse auf den folgenden Datensätzen verbesserte. Alle Modelle wurden einheitlich auf den drei Datensätzen trainiert. Als Hyperparameter wurden folgende Werte genutzt:

Hyperparameter	SQuAD -De	German QuAD	Winfy QuAD
Sequence length	384	384	384
Stride	128	128	128
Learning rate decay	Linear	Linear	Linear
Warmup steps	500	500	500
Learn rate	3e-5	3e-5	3e-5
Total Size (num. of entries)	89.996	13.722	1.746
Epochs	2	2	6
Batch size	24	24	24

Tabelle 3: Hyperparameter EQA-Feinabstimmung

Die Feinabstimmung dauerte aufgrund des deutlich geringeren Trainingsumfangs im Vergleich zum Vorab-Training für jedes Modell nur 30 Minuten.

Da die Performance des Modells in der Pre-Training-Aufgabe nicht automatisch Rückschlüsse auf die Leistung im Downstream-Task zulässt, wurde während des Vorab-Trainings alle 50.000 Schritte ein Sicherungspunkt des Modellzustands erstellt, um später jeden einzelnen Sicherungspunkt betrachten zu können. Die Feinabstimmung wurde für jeden dieser Sicherungspunkte durchgeführt, um den besten Zustand des Modells für die EQA-Aufgabe zu finden.

Abbildung 5: Winfy-ELECTRA-Base Performance aller Sicherungspunkte

In der Grafik sind der EM- und der F1-Wert auf dem WinfyQuAD für jeden Sicherungspunkt zu sehen. Es zeigte sich, dass das Modell beim Trainingsschritt 750.000 den höchsten EM-Wert erreicht.

AUSWERTUNG UND VERGLEICH

Die Feinabstimmung wurde für alle Modelle durchgeführt, und die Ergebnisse wurden dokumentiert.

	SQuAD-De-Test		**GermanQuAD-Test**		**WinfyQuAD-Test**	
Modell	EM	F1	EM	F1	EM	F1
deepset/gbert-base	50,35	65,65	58,21	75,61	45,80	64,79
deepset/gelectra-base	**53,23**	68,51	**62,93**	79,43	55,34	68,75
Winfy-ELECTRA-Base	52,91	**68,63**	**62,93**	**80,70**	**60,68**	**73,13**

Tabelle 4: EQA-Vergleich der Modelle (Angaben in %)

Es zeigte sich, dass Winfy-ELECTRA-Base sowohl auf dem GermanQuAD als auch auf dem WinfyQuAD am besten abschnitt. Auf dem GermanQuAD waren die Unterschiede nur marginal, während auf dem WinfyQuAD eine deutlichere Verbesserung zu verzeichnen war. Das deepset/gbert-base performt auf dem SQuAD-De und dem GermanQuAD ähnlich gut wie die anderen Modelle, ist allerdings deutlich schwächer auf dem WinfyQuAD. Dies kann darauf zurückzuführen sein, dass es während des Vorab-Trainings mit den wenigsten Daten trainiert wurde und somit vermutlich nicht mit Texten aus dem Hochschulkontext in Berührung gekommen ist da die verwendeten Datensätze nur „Wikipedia“ „OpenLegalData“ und „News“ waren [22]. Das deepset/gelectra-base dagegen wurde mit 163,4 GB an Daten trainiert, von denen ein Großteil (89 %) aus dem OSCAR-Datensatz bestand [20]. Somit liegt es nahe, dass unter diesen Daten auch Hochschultexte enthalten waren. Dabei handelt es sich aber lediglich um eine Vermutung, da es sich nicht überprüfen lässt.

INTEGRATION

Abschließend wurde das Winfy-ELECTRA-Base, welches auf das EQA feinabgestimmt wurde, in den Winfy-Chatbot eingebaut. Dadurch, dass das Training mithilfe der Transformers-Bibliothek von Huggingface durchgeführt wurde, lässt sich das Modell sehr einfach implementieren.

```
1.  from transformers import pipeline, AutoTokenizer, AutoModelForQuestionAnswering
2.
3.  model_path = "winfy-electra-base"
4.  tokenizer = AutoTokenizer.from_pretrained(model_path)
5.  model = AutoModelForQuestionAnswering.from_pretrained(model_path)
6.
7.  qa_pipeline = pipeline(
8.      "question-answering",
9.      model=model,
10.     tokenizer=tokenizer
11. )
12.
13. context = "Spätestens am letzten Tag der Bearbeitungsfrist ist eine digitale pdf-
    Version der Abschlussarbeit an das MND-Dekanat zur Fristwahrung zu mailen..."
14. question = "Wann ist spätestens die Abschlussarbeit abzugeben?"
15.
16. result = qa_pipeline(context=context, question=question, max_length=384)
```

Listing 3: Winfy-ELECTRA-Base EQA mit Huggingface

Als Rückgabewert erhält man die Anfangs- und Endposition der Antwort innerhalb des Kontexts, die extrahierte Antwortspanne als Text sowie einen Score, wie sicher das Modell sich mit der Vorhersage ist.

Es wurde beibehalten, dass zuerst der Antwortblock mithilfe des „German-Semantic"-Modells durch eine semantische Ähnlichkeitsüberprüfung vorausgewählt wird. Dieser Antwortblock wird nun zusammen mit der Frage an das EQA-Modell übergeben. Dies wird aus Performancegründen getan. Übergibt man alle Antwortblöcke gemeinsam mit der gestellten Frage an das EQA-Modell, so beträgt die Verarbeitungszeit auf einer CPU vier Minuten, während der eben beschriebene Ansatz nur zwei Sekunden dauert.

Das EQA-Modell gibt die exakte Position der Antwort zurück. Diese Information wird genutzt, um den Absatz zu bestimmen, der die Antwort enthält. Dieser wird an den Endnutzer ausgegeben (siehe Anhang).

FAZIT UND AUSBLICK

Es zeigte sich, dass das domänenspezifische Pre-Training seinen gewünschten Zweck erfüllen konnte. Das Winfy-ELECTRA-Base erzielte die besten Werte auf dem EQA-Hochschuldatensatz. Weiterhin konnte durch die Integration des neuen EQA-Modells die Antwortgenauigkeit des Chatbots verbessert und somit das Ziel der Arbeit erreicht werden.

Es wäre zudem interessant, das Modell mit mehr Parametern zu trainieren, um weitere mögliche Verbesserungen zu untersuchen. Auch könnte ein fortschrittlicherer Architekturansatz, wie zum Beispiel DeBERTaV3 [21], verfolgt werden.

Aufgrund der für das Training verwendeten EQA-Datensätze versucht das Modell, immer eine Antwort zu finden. Lässt sich die Frage nicht anhand des Kontexts beantworten, würde das Modell dennoch eine Textspanne ausgeben.

Dieses Problem wurde im SQuAD v.2.0 angegangen. Dieser ist allerdings zum Zeitpunkt der Projektdurchführung nur in englischer Sprache verfügbar. Sobald er auch in deutscher Sprache zugänglich ist, wäre es interessant, das Modell darauf zu trainieren.

LITERATUR

[1] Beltagy, Iz; Lo, Kyle; Cohan, Arman: SciBERT: A Pretrained Language Model for Scientific Text, o.O., 2019, DOI: https://doi.org/10.48550/arXiv.1903.10676

[2] Gu, Yu; Tinn, Robert; Cheng, Hao; Lucas, Michael; Usuyama, Naoto; Liu, Xiaodong; Naumann, Tristan; Gao, Jianfeng; Poon, Hoifung: Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing, in: ACM Transactions on Computing for Healthcare vol. 3 no. 1, 2021, S. 1-23, DOI: https://doi.org/10.1145/3458754

[3] Tomar, Sahaj: Sahajtomar/German-semantic, huggingface, o.O., o.J., online im Internet: URL: https://huggingface.co/Sahajtomar/German-semantic [Stand 19.12.2024]

[4] Ritz, Harald; Tansel, Dogus: Entwicklung eines KI-basierten FAQ-Chatbots für die Hochschule im Bereich Prüfungsangelegenheiten, in: Anwendungen und Konzepte in der Wirtschaftsinformatik (AKWI) Nr. 17 (2023), S.81-92, DOI: https://doi.org/10.26034/lu.akwi.2023.n17

[5] Wikipedia, o.O., o.J., online im Internet: URL: https://www.wikipedia.org/ [Stand 02.01.2025]

[6] Wikivoyage, o.O., o.J., online im Internet: URL: https://www.wikivoyage.org/ [Stand 02.01.2025]

[7] May, Philip; Reißel, Philipp: GC4 Corpus - The German colossal, cleaned Common Crawl corpus, o.O., 2021, online im Internet: URL: https://german-nlp-group.github.io/projects/gc4-corpus.html [Stand 25.12.2024]

[8] CommonCrawl - Common Crawl maintains a free, open repository of web crawl data that can be used by anyone, o.O., o.J., online im Internet: URL: https://commoncrawl.org/ [Stand 02.01.2025]

[9] Hiemstra, Djoerd: A probabilistic justification for using tf×idf term weighting in information retrieval, in: Int J Digit Libr 3, 2000, S. 131-139, DOI: https://doi.org/10.1007/s007999900025

[10] Proisl, Thomas; Uhrig, Peter; Cook, Paul; Evert, Stefan; Schäfer, Roland; Stemle, Egon (Hrsg.): SoMaJo: State-of-the-art tokenization for German web and social media texts, in: Proceedings of the 10th Web as Corpus Workshop, Berlin, 2016, S. 57-62, DOI: https://doi.org/10.18653/v1/W16-2607

[11] Wenzek, Guillaume; Lachaux, Marie-Anne; Conneau, Alexis; Chaudhary, Vishrav; Guzmán, Francisco; Joulin, Armand; Grave,

Edouard: CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data, o.O., 2019, DOI: https://doi.org/10.48550/arXiv.1911.00359

[12] Lewis, Patrick; Oguz, Barlas; Rinott, Ruty; Riedel, Sebastian; Schwenk, Holger; Jurafsky, Dan; Chai, Joyce; Schluter, Natalie; Tetreault, Joel (Hrsg.): MLQA: Evaluating Cross-lingual Extractive Question Answering, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA, 2020, S. 7315–7330, DOI: https://doi.org/10.18653/v1/2020.acl-main.653

[13] Möller, Timo; Risch, Julian; Pietsch, Malte: GermanQuAD and GermanDPR: Improving Non-English Question Answering and Passage Retrieval, in: Proceedings of the 3rd Workshop on Machine Reading for Question Answering, Punta Cana, Dominikanische Republik, 2021, S. 42-50, DOI: https://doi.org/10.18653/v1/2021.mrqa-1.4

[14] Tkachenko, Maxim; Malyuk, Mikhail; Holmanyuk, Andrey; Liubimov, Nikolai: Label Studio: Data labeling software, o.O., 2020, online im Internet: URL: https://github.com/HumanSignal/label-studio [Stand 20.12.2024]

[15] Clark, Kevin; Luong, Minh-Thang; Le, V. Quoc; Manning, Christopher D.: ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, o.O., 2020, DOI: https://doi.org/10.48550/arXiv.2003.10555

[16] Wolf, Thomas; Debut, Lysandre; Sanh, Victor; Chaumond, Julien; Delangue, Clement; Moi, Anthony; Cistac, Pierric; Rault, Tim; Rémi, Louf; Funtowicz, Morgan; Davison, Joe; Shleifer, Sam; von Platen, Patrick; Ma, Clara; Jernite, Yacine; Plu, Julien; Xu, Canwen; Le Scao, Teven; Gugger, Sylvain; Drame, Mariama; Lhoest, Quentin; Rush, Alexander M.: HuggingFace's Transformers: State-of-the-art Natural Language Processing, o.O., 2020, DOI: https://doi.org/10.48550/arXiv.1910.03771

[17] Wang, Luqi; Zheng, Kaiwen; Qian, Liyin; Li, Sheng: A Survey of Extractive Question Answering, in: 2022 International Conference on High Performance Big Data and Intelligent Systems (HDIS), Tianjin China, 2022, S. 147-153, DOI: https://doi.org/10.1109/HDIS56859.2022.9991478

[18] Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of NAACL-HLT 2019, Minneapolis, Minnesota, 2019, S. 4171-4186, DOI: https://doi.org/10.18653/v1/N19-1423

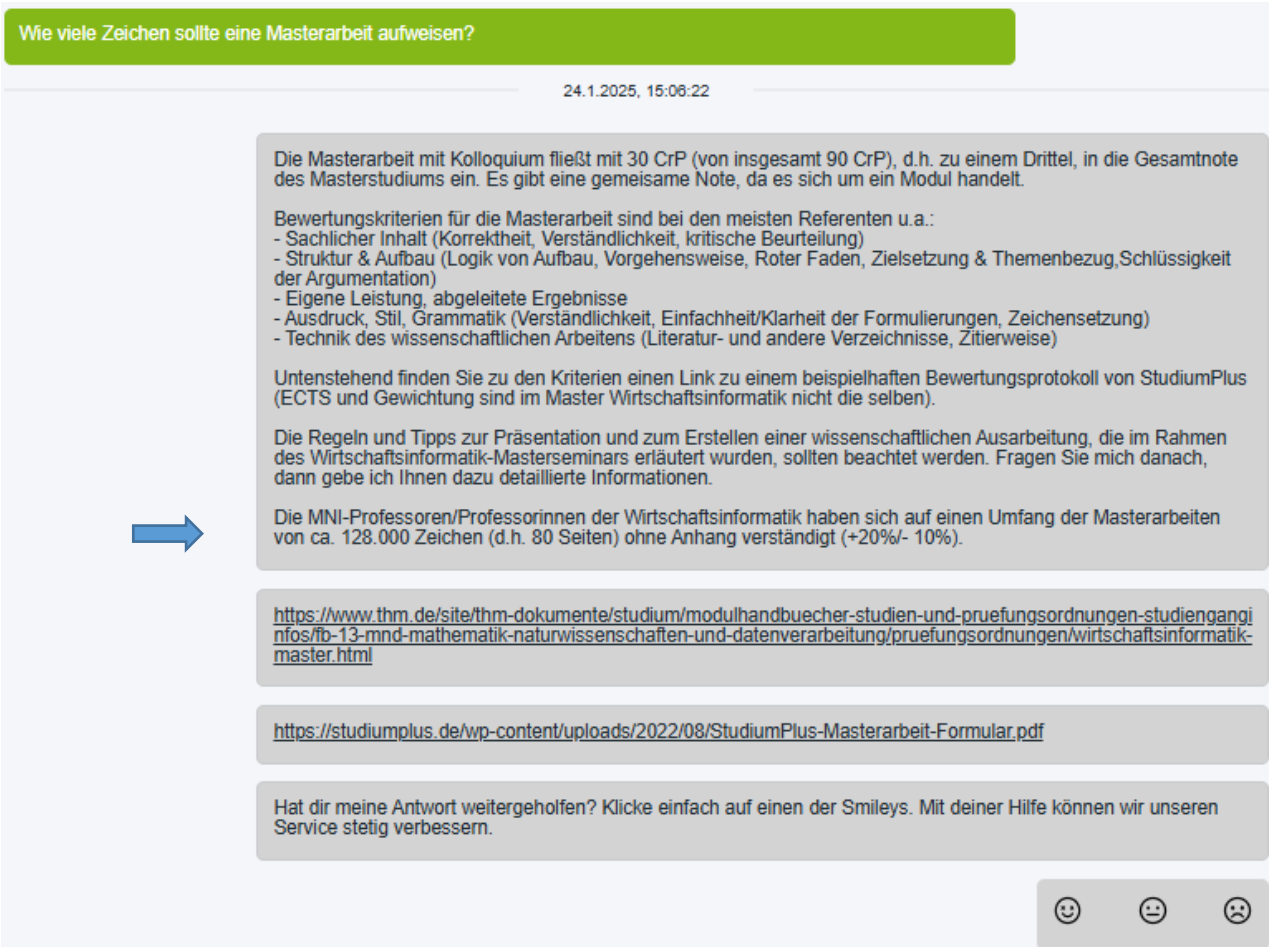
[19] Rajpurkar, Pranav; Zhang, Jian; Lopyrev, Konstantin; Liang, Percy: SQuAD: 100,000+ Questions for Machine Comprehension of Text, o.O., 2016, DOI: https://doi.org/10.48550/arXiv.1606.05250

[20] Chan, Branden; Schweter, Stefan; Möller, Timo; Scott, Donia; Bel, Nuria; Zong, Chengqing (Hrsg.): German's Next Language Model, in: Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spanien, 2020, S. 6788–6796, DOI: https://doi.org/10.18653/v1/2020.coling-main.598

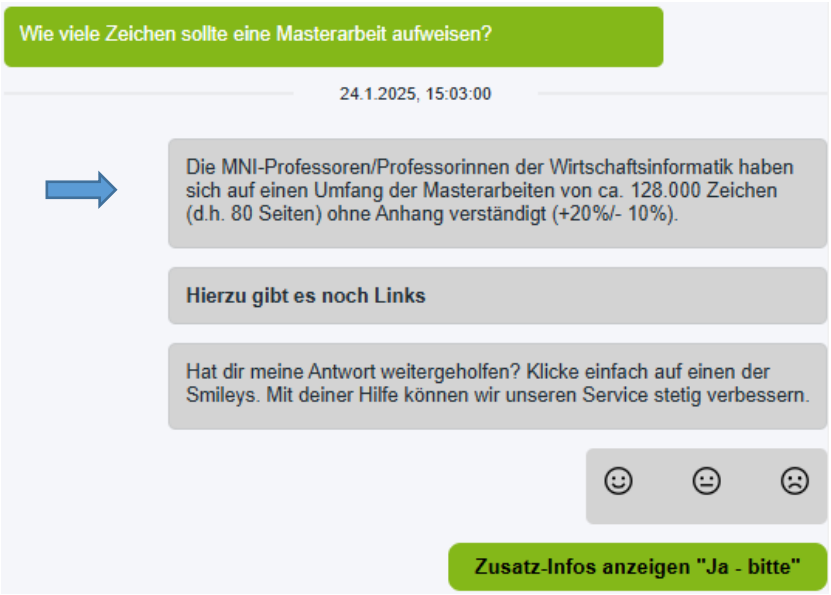
[21] He, Pengcheng; Gao, Jianfeng; Chen, Weizhu: DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing, o.O., 2023, DOI: https://doi.org/10.48550/arXiv.2111.09543

[22] Chan, Branden; Möller, Timo; Pietsch, Malte; Soni, Tanay: German BERT, o.O., 2019, online im Internet: URL: https://huggingface.co/google-bert/bert-base-german-cased [Stand: 30.12.2024]

ANHANG



Anhang 1: Antwortumfang vorher



Anhang 2: Antwortumfang nachher

Eine empirische Untersuchung zur Erkennung und ethischen Einschätzung KI-generierter Bilder sowie zur Erstellung und Verbreitung sensibler Inhalte über KI-basierte Bildgenerierungstools

Leon Hobelmann

Hochschule für Technik und Wirtschaft Berlin
Wirtschaftsinformatik
Treskowallee 8
10318 Berlin
E-Mail:
mail@leon-hobelmann.de

Birte Malzahn

Hochschule für Technik und Wirtschaft Berlin
Wirtschaftsinformatik
Treskowallee 8
10318 Berli0n
E-Mail:
birte.malzahn@htw-berlin.de

Schlüsselwörter

Digitale Ethik, Midjourney, KI-Bildgenerierung, Desinformation, Plattformrichtlinien

Zusammenfassung

KI-basierte Bildgeneratoren wie Midjourney haben sich als feste Werkzeuge in Wirtschaft und Bildung etabliert. Zu den Vorteilen zählen die Unterstützung von Lern- und Kreativprozessen sowie die Erweiterung gestalterischer Möglichkeiten (Bendel 2025). Mit der zunehmenden Verbreitung dieser Werkzeuge rücken jedoch auch ethische Fragestellungen in den Fokus: KI-generierte Bilder können zur Verbreitung von Desinformation und zur gezielten Beeinflussung öffentlicher Wahrnehmung eingesetzt werden. Des Weiteren kann bei der Generierung von KI-Bildern das Urheberrecht verletzt werden, wenn zum Training der KI geschützte Inhalte verwendet werden (Bird et al. 2023). Plattformrichtlinien und technische Filter adressieren diese Risiken nur begrenzt, da problematische Inhalte trotz Beschränkungen erzeugt oder bzw. untersagte Inhalte mittels Umgehungsstrategien realisiert werden können (Leow 2023).

Vor diesem Hintergrund untersuchte diese Bachelorarbeit empirisch die Erkennung und ethische Bewertung KI-generierter Bildinhalte. Des Weiteren wurde anhand von verfügbaren KI-generierten Bildern auf der Plattform Midjourney analysiert, inwieweit Verstöße gegen die eigenen Richtlinien der Plattform vorliegen (Hobelmann 2025). Der theoretische Teil der Arbeit berücksichtigte KI-ethische Prinzipien wie Transparenz, Verantwortung, Nichtschadensgebot und Fairness, die in KI-Ethikleitlinien als zentrale normative Bezugspunkte hervorgehoben werden (Jobin et al. 2019). Ergänzend wurde der EU AI Act als rechtlicher Rahmen herangezogen, der Transparenzpflichten für KI-Systeme und ein risikobasiertes Regulierungskonzept vorsieht (Europäische Union 2024).

Die empirische Untersuchung wurde mithilfe eines Online-Fragebogens durchgeführt. Der Fragebogen wurde am 28.06.2025 veröffentlicht und im eigenen Umfeld verteilt. Insgesamt wurden 87 verwertbare Datensätze ausgewertet. Die Stichprobe umfasste 52 männliche, 24 weibliche und eine diverse Person; die 23- bis 27-jährigen bildeten mit etwa 33 % die größte Altersgruppe. In einem Three-Alternative-Forced-Choice-Design wurden Bildersets vorgelegt, die jeweils aus drei sehr ähnlichen Darstellungen bestanden, von denen jeweils eine reale Fotografie und zwei mit Midjourney selbst-generierte Bilder waren. Die Teilnehmenden sollten in jedem Set die reale Fotografie auswählen. Ziel war es, die Erkennungsfähigkeit synthetischer Bilder zu erfassen.

Abbildung 1 - Beispiel aus der 3 AFC Aufgabe (Quelle linkes Bild: Adbullah, 2022)

Anschließend bewerteten die Befragten auf Likert-Skalen die ethische Vertretbarkeit von KI-generierten Produktdarstellungen, Karikaturen (z. B. Queen Elisabeth II

auf einem Skateboard), Vorher-Nachher-Bilder (z. B. eine zweiteilige Frontalaufnahme derselben Person: höherer Körperfettanteil vs. deutlich erhöhte Muskelmasse) sowie von Motiven mit potenziell sensiblen Inhalten (konfliktbezogene bzw. körpernahe Darstellungen). Die Ergebnisse zeigen, dass die Teilnehmenden das reale Foto in den Bildersets im Mittel nur in etwa 47 % der Fälle korrekt identifizierten und damit zwar signifikant über dem Zufallsniveau von 33,3 %, aber deutlich unter einer zuverlässigen Erkennungsleistung liegen; signifikante Unterschiede zwischen Altersgruppen und Geschlechtern traten hierbei nicht auf. In der anschließenden Bewertung der Bildkategorien wurden KI-generierte Produktdarstellungen überwiegend als ethisch vertretbar eingestuft. Motive mit Gewalt- bzw. erotischem Bezug erhielten die niedrigsten Zustimmungswerte. Über alle Kategorien hinweg bewerteten männliche Teilnehmende die gezeigten Inhalte signifikant positiver als weibliche. Zugleich war mit steigendem Alter eine tendenziell strengere Beurteilung erkennbar.

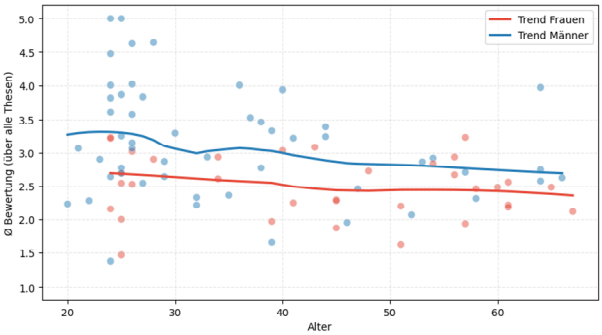


Abbildung 2 - Durchschnittliche Bewertung der Bildinhalte nach Alter und Geschlecht

Anschließend wurden auf der Plattform Midjourney verfügbare Bildbeispiele identifiziert, die auf Grundlage der geltenden Nutzungsbedingungen und Inhaltsrichtlinien als kritisch einzustufen sind (Midjourney o.J.). Die Einordnung der Befunde orientierte sich an risikobasierten Kategorisierungsansätzen für generative Modelle, wie sie in der „Topology of Risk“ beschrieben werden (Bird et al. 2023). Die qualitative Analyse ergab, dass problematische Inhalte trotz bestehender Richtlinien generiert und verbreitet werden, dass einzelne Inhalte gegen die Plattformvorgaben verstoßen und dass Nutzer*innen wiederkehrende Strategien zur Umgehung der Filter einsetzen, was auf substanzielle technische Lücken in der Durchsetzung der Richtlinien des Anbieters hinweist. Es zeigt sich, dass realitätsnahe synthetische Bilder mit geringem technischem Aufwand erzeugt werden können und ein erhebliches Potenzial für Desinformation, politische Einflussnahme und personenbezogene Rufschädigung bergen.

Die Zusammenführung beider Teilstudien zeigt, dass KI-Bilder durch Nutzer*innen nicht zuverlässig erkannt werden und KI-generierte sensible Motive von den Befragten überwiegend als ethisch problematisch bewertet werden. Zugleich sind vergleichbare Inhalte auf Plattformen verfügbar und teils trotz Verbot erzeugbar. Daraus ergibt sich konkreter Handlungsbedarf: Plattformrichtlinien müssen konsequent durchgesetzt werden, Filter sollten kontextsensitive Prüfungen unterstützen, Kennzeichnung und Herkunftsnachweise im Sinne des EU AI Acts sind verbindlich umzusetzen.

Literatur

Abdullah, Sami (2022), *Foto von Sami Abdullah auf Pexels*, Pexels https://www.pexels.com/de-de/foto/strasse-auto-vintage-mercedes-13818893/ (letzter Zugriff am 14.11.2025)

Bendel, Oliver (2025), *Image Synthesis from an Ethical Perspective,* AI & SOCIETY, 2025, Band 40, Auflage 2, https://doi.org/10.1007/s00146-023-01780-4

Bird, Charlotte, Ungless, Eddie L. und Kasirzadeh, Atoosa (2023) *Topology of Risks of Generative Text-to-Image Models*, in: AIES 23: Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society, Association for Computing Machinery, Montreal https://doi.org/10.48550/arXiv.2307.05543

Europäische Union (2024) *Verordnung über künstliche Intelligenz,* EUR-Lex https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32024R1689 (letzter Zugriff am 14.11.2025)

Hobelmann, Leon (2025) *Ethische Herausforderungen in der Bildgenerierung durch Künstliche Intelligenz am Beispiel von Midjourney,* Bachelorarbeit, Hochschule für Technik und Wirtschaft Berlin

Jobin, Anna, Ienca, Marcello und Vayena, Effy (2019) *The Global Landscape of AI Ethics Guidelines,* Nature Machine Intelligence, 2019, Band 1, Aufl. 9, S. 389–399 https://doi.org/10.1038/s42256-019-0088-2

Leow, Mikelle (2023) *Midjourney Bans Images Of China's Xi Jinping, Warns Users Not To Get Sneaky,* Designtaxi, URL: https://designtaxi.com/news/422929/Midjourney-Bans-Images-Of-Chinas-Xi-Jinping-Warns-Users-Not-To-GetSneaky/ (letzter Zugriff am 14.11.2025)

Midjourney (o.J.) *Community Guidelines,* Midjourney https://docs.midjourney.com/hc/en-us/articles/32013696484109-Community-Guidelines (letzter Zugriff am 14.11.2025)

Evaluierung der Einsatzmöglichkeiten von Process Mining als Analyseinstrument für die Logistik eines Automobilherstellers

Arno Müller
Hochschule Pforzheim
Tiefenbronner Straße 65
75175 Pforzheim
arno.mueller95@gmail.com

Frank Morelli
Hochschule Pforzheim
Tiefenbronner Straße 65
75175 Pforzheim
frank.morelli@hs-pforzheim.de

ABSTRACT

Diese Arbeit evaluiert Process Mining als ergänzendes Informationsinstrument für die Produktionslogistik eines Automobilherstellers, spezifisch für den innerbetrieblichen Transport von Presswerk und Rohbau. Angesichts der Nachteile traditioneller KPI-Systeme wird das Potenzial von Process Mining zur Schaffung objektiver Prozesstransparenz untersucht. Die Untersuchung nutzt ein sequenzielles Mixed-Methods-Design. Zunächst wurden in einer qualitativen Phase Experteninterviews geführt, um die Bewertungskriterien für ein Informationssystem zu definieren. Darauf aufbauend folgte eine Fallstudie in Form eines Process Mining Prototyps. Parallel dazu wurde ein quantitatives Bewertungsmodell mittels Nutzwertanalyse konzipiert, um im letzten Schritt einen systematischen Vergleich beider Systeme durchzuführen. Die evaluative Haupterkenntnis aus der Nutzwertanalyse war, dass das etablierte KPI-System den Process Mining Prototyp aufgrund dessen geringen Reifegrads, beispielsweise der manuellen Datenextraktion, noch übertrifft. Aus diesen Ergebnissen konnte jedoch das überlegene strategische Potenzial von Process Mining abgeleitet werden. Eine Szenarioanalyse belegte, dass eine voll integrierte Process Mining Lösung dem bestehenden System klar überlegen wäre. Der strategische Mehrwert hängt somit entscheidend von einer robusten technischen Integration und der organisatorischen Verankerung ab.

SCHLÜSSELWÖRTER

Process Mining, Automobilindustrie, Logistik, Kennzahlen, Mixed-Methods

EINLEITUNG UND PROBLEMSTELLUNG

Die Automobilindustrie agiert in einem Umfeld, das von exzessiver Produktvielfalt, globalisierten Liefernetzwerken und volatilen Märkten geprägt ist. Diese Komplexität stellt insbesondere die Produktionslogistik vor fundamentale Steuerungsprobleme. Der Trend zur *Mass-Customization*, der Fähigkeit, individuelle Produkte zu Kosten der Massenfertigung herzustellen, führt zu einer nahezu unbegrenzte Variantenvielfalt, die den Materialfluss hochgradig komplex gestaltet. Zur Steuerung dieser Prozesse werden traditionell aggregierte Kennzahlensysteme (KPIs) eingesetzt. Diese stoßen jedoch an inhärente Limitationen:

- **Retrospektive Sicht:** KPIs quantifizieren meist nur das Ergebnis eines Prozesses (deskriptive Dimension), nicht aber den Prozess selbst.
- **Fehlende Diagnosefähigkeit:** Die zugrundeliegenden prozessualen Abläufe und kausalen Treiber (diagnostische Dimension) bleiben verborgen.
- **Subjektivität:** Die Interpretation der KPIs erfordert hohe Prozessexpertise, was die Entscheidungsfindung subjektivitätsanfällig macht.

Hier manifestiert sich eine methodische Lücke: Es fehlt an Instrumenten, die eine durchgängige, datengestützte und objektive Transparenz über die tatsächlichen End-to-End-Materialflüsse schaffen. Process Mining (PM) positioniert sich als Technologie, um diese Lücke zu schließen, indem es reale Prozessabläufe aus digitalen Spuren in IT-Systemen (z.B. SAP) rekonstruiert.

ZIELSETZUNG

Das zentrale Ziel dieser Arbeit war die systematische Untersuchung, welchen konkreten Mehrwert Process Mining als ergänzendes Informationsinstrument für die Produktionslogistik eines Automobilherstellers generieren kann. Dies wurde im spezifischen Fall des innerbetrieblichen Transports für Presswerk und Rohbau in einem Werk untersucht. Es sollte empirisch validiert werden, ob die Methode geeignet ist, die bestehende Intransparenz in den Materialflüssen zu schließen. Die primäre Forschungsfrage lautete daher:

„Eignet sich Process Mining als ergänzendes Informationsinstrument für die Produktionslogistik in diesem Anwendungsfall?“

METHODISCHES VORGEHEN

Zur Beantwortung der Forschungsfrage wurde ein sequenziell-exploratives Mixed-Methods-Design nach Creswell & Creswell (2023) angewendet. Dieses Vorgehen war notwendig, da kein etabliertes Bewertungsinstrument für Process Mining im spezifischen Kontext vorlag und die Kriterien empirisch entwickelt werden mussten. Die Untersuchung gliederte sich in drei Phasen,

deren Ablauf und Zusammenhänge in Abbildung 1 visualisiert sind. Die folgenden Abschnitte beschreiben diese Phasen im Detail.

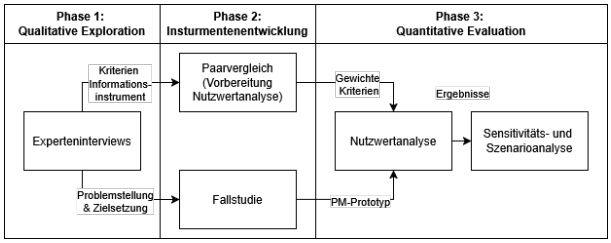


Abbildung 1: Methodisches Vorgehen

Phase 1: Qualitative Exploration

Zunächst wurden acht **leitfadengestützte Experteninterviews** mit Fach- und Führungskräften der Logistik des Automobilherstellers sowie kooperierender Dienstleister geführt. Ziel war es, die Ausgangssituation, bestehende Problemstellungen und die Anforderungen an ein Informationssystem zu explorieren. Die Auswertung der Transkripte erfolgte mittels einer **zusammenfassenden qualitativen Inhaltsanalyse** nach Mayring (2015). Ein zentrales Ergebnis dieser Phase war die empirische Ableitung der **sechzehn Bewertungskriterien**, die als Grundlage für die spätere Evaluation dienten.

Phase 2: Fallstudie & Instrumentenentwicklung

Die zweite Phase der Untersuchung war zweigeteilt und umfasste sowohl die Konzeption des Bewertungsinstruments als auch die praktische Durchführung der Fallstudie.

Instrumentenentwicklung
Für die Instrumentenentwicklung wurde eine **Nutzwertanalyse** konzipiert. Die dafür notwendigen sechzehn Bewertungskriterien, die in der ersten Phase aus den Experteninterviews abgeleitet wurden, mussten objektiv gewichtet werden. Dies erfolgte durch einen systematischen **Paarvergleich** nach der AHP-Methode, wie er exemplarisch in Abbildung 2 dargestellt ist. Der Expertenkreis musste hierbei für jedes Kriterienpaar entscheiden, welches als wichtiger erachtet wird.

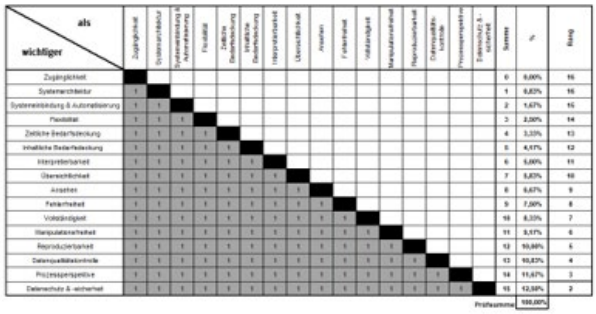


Abbildung 2: Paarvergleich

Aus der Aggregation dieser paarweisen Vergleiche resultierte die finale Gewichtung der sechzehn Kriterien, die in Abbildung 3 aufgeführt ist. Dieser Prozess stellte eine objektive Priorisierung sicher und zeigte, dass Kriterien wie Datenschutz und -sicherheit sowie Fehlerfreiheit die höchste Relevanz für die Anwender besaßen.

Kriterien	Gewichtung	Rang
Datenschutz und -sicherheit	12,29%	1
Fehlerfreiheit	11,78%	2
Manipulationsfreiheit	10,64%	3
Inhaltliche Bedarfsdeckung	9,92%	4
Vollständigkeit	8,26%	5
Datenqualitätskontrolle	8,26%	5
Interpretierbarkeit	7,44%	7
Zeitliche Bedarfsdeckung	6,61%	8
Übersichtlichkeit	6,61%	8
Flexibilität	4,96%	10
Prozessperspektive	4,13%	11
Zugänglichkeit	3,31%	12
Systemeinbindung und Automatisierung	2,48%	13
Systemarchitektur	1,65%	14
Ansehen	0,83%	15
Reproduzierbarkeit	0,83%	15

Abbildung 3: Gewichtete Kriterien

Fallstudie
Parallel dazu wurde die **prototypische Process Mining Lösung** als Fallstudie implementiert. Das Vorgehen orientierte sich konsequent an der PM-Quadrat-Methodik von van Eck und Kollegen (2015), welche alle sechs Phasen eines Process Mining Projekts abdeckt.

Die **erste Phase** der Planung definierte den Untersuchungsfokus auf die Materialflüsse von Presswerk und Rohbau. In der **zweiten Phase**, der Extraktion, erfolgte der manuelle Export der relevanten SAP-Transaktionsdaten.

Die **dritte Phase**, die Datenaufbereitung, stellte die größte methodische Herausforderung dar. Die SAP-Rohdaten lagen in einer transaktionalen „Von-Nach"-Struktur vor und waren für ein traditionelles, fallzentriertes Process Mining ungeeignet. Das Kernproblem war, dass die Handling Units als Mehrwegladungsträger theoretisch unendliche Lebenszyklen aufweisen, was eine valide Prozessanalyse verunmöglichte. Diese Hürde wurde durch einen mehrstufigen ETL-Transformationsprozess gelöst. Zuerst erfolgte eine Atomisierung, bei der jede „Von-Nach"-Buchung in zwei separate Ereignisse, eine „Auslagerung" und eine „Einlagerung", aufgespalten wurde. Darauf folgte die Segmentierung, bei der der kontinuierliche Lebenszyklus jeder Handling Unit anhand von Geschäftsregeln in logisch abgeschlossene Prozessinstanzen, sogenannte Zyklen, unterteilt wurde. Erst diese neu geschaffene „HU-Zyklus-ID" diente als valider „Case".

In der **vierten Phase**, dem Mining und der Analyse, wurde dieser aufbereitete Datensatz genutzt, um mittels Process Discovery die realen Ist-Prozesse zu rekonstruieren, mittels Conformance Checking exemplarisch Abwechungen zu prüfen und mittels Enhancement die Auswirkungen von Ineffizienzen auf die Durchlaufzeit zu quantifizieren.

Die **fünfte Phase** der Evaluierung validierte die Analyseergebnisse intern. Die Methodik schloss mit der **sechsten Phase**, der Implementierung, ab, welche die Erkenntnisse in konkrete Handlungsempfehlungen übersetzte.

Phase 3: Quantitative Evaluation

In der finalen, dritten Forschungsphase erfolgte die quantitative Evaluation durch die Anwendung des in Phase zwei entwickelten Instruments. Die **Nutzwertanalyse** nach Zangemeister (2014) bildete den methodischen Kern, um die beiden Alternativen, das etablierte KPI-System und den PM-Prototyp, systematisch zu vergleichen. Die Bewertung der sechzehn Kriterien erfolgte nach einem zweigeteilten Ansatz. Kriterien, die die subjektive Nutzerwahrnehmung betreffen, wie beispielsweise die Zugänglichkeit oder Übersichtlichkeit, wurden durch den Expertenkreis beurteilt. Objektiv-technische Kriterien, wie die Systemarchitektur oder der Automatisierungsgrad, wurden vom Verfasser auf Basis der technischen Implementierung und Analyse bewertet.

Der Gesamtnutzwert einer Alternative N_{alt} berechnet sich dabei als Summe der gewichteten Einzelerfüllungsgrade der jeweiligen Kriterien i nach der Formel:

$$N_{alt} = \sum_{i=1}^{n} G_i \cdot B_i$$

wobei G_i die Gewichtung und B_i die Bewertung des Kriteriums i ist. Die finalen Ergebnisse dieser Bewertung sind in Abbildung 4 zusammengefasst und werden im folgenden Kapitel detailliert diskutiert.

		PM-Lösung		KPI (SFM)		
Nutzwertanalyse	Gewichtung	Bewertung	Wert	Bewertung	Wert	Differenz (PM-SFM)
Zugänglichkeit	3,31%	2	0,07	3	0,10	-1
Systemarchitektur	1,65%	3	0,05	1	0,02	2
Systemeinbind. & Automatisierung	2,48%	1	0,02	3	0,07	-2
Flexibilität	4,96%	2	0,10	1	0,05	1
Zeitliche Bedarfsdeckung	6,61%	1	0,07	2	0,13	-1
Inhaltliche Bedarfsdeckung	9,92%	2	0,20	1	0,10	1
Interpretierbarkeit	7,44%	3	0,22	2	0,15	1
Übersichtlichkeit	6,61%	2	0,13	3	0,20	-1
Ansehen	0,83%	1	0,01	2	0,02	-1
Fehlerfreiheit	11,57%	1	0,12	3	0,35	-2
Vollständigkeit	8,26%	1	0,08	3	0,25	-2
Manipulations-freiheit	10,74%	3	0,32	3	0,32	0
Reproduzier-barkeit	0,83%	3	0,02	3	0,02	0
Datenqualitäts-kontrolle	8,26%	2	0,17	1	0,08	1
Prozessperspek-tive	4,13%	2	0,08	1	0,04	1
Datenschutz und -sicherheit	12,40%	3	0,37	3	0,37	0
Summe			**2,03**		**2,27**	-0,24

Abbildung 4: Nutzwertanalyse

Um die Stabilität dieser Ergebnisse zu prüfen, wurde zudem eine Robustheitsanalyse durchgeführt. Eine **Sensitivitätsanalyse**, deren Ergebnisse in Abbildung 5 dargestellt sind, prüfte, wie stark die einzelnen Kriterien das Endergebnis beeinflussen. Sie visualisiert die gewichtete Bewertungsdifferenz für jedes Kriterium und identifiziert so die Haupttreiber für den Gesamtnutzwertunterschied.

Abbildung 5: Tornado-Diagramm zur Sensitivitätsanalyse

Zusätzlich validierte eine **Szenarioanalyse** das Zukunftspotenzial des Process Mining Prototyps. Hierfür wurden die Bewertungen des PM-Prototyps in den kritischen, reifegradbedingten Kriterien, wie Systemeinbindung und Fehlerfreiheit, hypothetisch auf einen vollintegrierten Zustand angehoben. Diese ganzheitliche Veränderung der Parameter simulierte die Auswirkungen einer vollständigen technischen Implementierung auf das Gesamtergebnis.

WESENTLICHE ERGEBNISSE

Die analytischen Kernnerkenntnisse aus der Fallstudie waren die Identifikation signifikanter Prozessineffizienzen. Der Prototyp deckte beispielsweise unnötige Umlagerungen und aufwändige Nacharbeitsschleifen auf. Die Enhancement-Analyse zeigte, dass diese Abwechungen die durchschnittliche Durchlaufzeit der betroffenen Handling Units teilweise mehr als verdoppelten.

Die evaluative Haupterkenntnis aus der Nutzwertanalyse, deren detaillierte Ergebnisse in Abbildung 4 zusammengefasst sind, war, dass das etablierte KPI-System den Process Mining Prototyp aktuell noch übertrifft.

Das etablierte System erreichte einen **Gesamtnutzwert** von **2,27**, während der PM-Prototyp einen Wert von **2,03** erzielte. Diese Diskrepanz ist nicht auf eine technologische Unterlegenheit von Process Mining zurückzuführen, sondern auf den geringen Reifegrad des Prototyps. Das KPI-System profitierte von seiner vollständigen technischen Integration, Automatisierung und der Nutzung der Datengrundgesamtheit, welche in der Bewertung hoch gewichtete Kriterien waren. Der PM-Prototyp erhielt Abzüge durch die manuelle Datenextraktion und die Nutzung einer limitierten Stichprobe.

Die Sensitivitätsanalyse, visualisiert in Abbildung 5, stützt diese Erkenntnis und verdeutlicht die Stabilität der Entscheidung.

Sie zeigt, dass die größten negativen Einflüsse auf die PM-Lösung von den hoch gewichteten Kriterien Fehlerfreiheit und Vollständigkeit ausgingen, deren Bewertungsdifferenz zugunsten des KPI-Systems signifikant war. Die positiven Aspekte des PM-Prototyps, wie die Prozessperspektive, fielen im direkten Vergleich weniger stark ins Gewicht.

Aus diesen Ergebnissen konnte jedoch das überlegene strategische Potenzial von Process Mining abgeleitet werden. Die Szenarioanalyse, welche die ganzheitliche Anhebung des Reifegrads auf ein vollintegriertes System simulierte, belegte, dass eine voll integrierte Process Mining Lösung einen Nutzwert von **2,57** erreichen und damit dem bestehenden System klar überlegen wäre.

FAZIT

Die Studie validiert die grundsätzliche Eignung von Process Mining als ergänzendes Informationsinstrument für die komplexe Intralogistik in der Automobilindustrie. Die Technologie schließt die methodische Lücke traditioneller Kennzahlensysteme, indem sie eine datengestützte, diagnostische Transparenz über reale End-to-End-Prozesse schafft.

Die Diskrepanz zwischen dem unterlegenen Nutzwert des Prototyps und dem überlegenen Potenzial einer vollintegrierten Lösung ist die zentrale Erkenntnis der Arbeit. Sie belegt, dass der strategische Mehrwert von Process Mining weniger von der Technologie selbst als von einer robusten technischen Integration, insbesondere einer automatisierten Datenanbindung, abhängt. Die Handlungsempfehlung lautet daher, das etablierte KPI-System kurzfristig beizubehalten, aber parallel die strategische, vollintegrierte Implementierung von Process Mining voranzutreiben, um mittelfristig die operative Steuerung auf eine überlegene datenbasierte Grundlage zu stellen.

LITERATUR

J. W. Creswell und J. D. Creswell, *Research design: qualitative, quantitative, and mixed methods approaches*, 6. Aufl. Los Angeles London New Delhi Singapore Washington DC Melbourne: Sage, 2023.

F. Klug, *Logistikmanagement in der Automobilindustrie: Grundlagen der Logistik Im Automobilbau*, 2. Aufl. in VDI-Buch Ser. Berlin, Heidelberg: Springer Berlin / Heidelberg, 2018.

W. M. P. van der Aalst, *Process Mining: Data Science in Action*, 2. Aufl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016.

P. Mayring, *Qualitative Inhaltsanalyse: Grundlagen und Techniken*, 12. Aufl. in Beltz Pädagogik. Weinheim: Beltz, 2015.

M. L. van Eck, X. Lu, S. J. J. Leemans, und W. M. P. van der Aalst, „PM^2: A Process Mining Project Methodology“, in *Advanced Information Systems Engineering*, J. Zdravkovic, M. Kirikova, und P. Johannesson, Hrsg., Cham: Springer International Publishing, 2015, S. 297–313.

R. Y. Wang und D. M. Strong, „Beyond Accuracy: What Data Quality Means to Data Consumers“, *J. Manag. Inf. Syst.*, Bd. 12, Nr. 4, S. 5–33, März 1996.

H. Werner, *Supply Chain Controlling: Grundlagen, Performance-Messung und Handlungsempfehlungen*, 2. Aufl. Wiesbaden: Springer Fachmedien Wiesbaden GmbH, 2022.

C. Zangemeister, *Nutzwertanalyse in der Systemtechnik: eine Methodik zur multidimensionalen Bewertung und Auswahl von Projektalternativen*, 5. Aufl. Winnemark: Zangemeister & Partner, 2014.

SAP Business Technology Platform als Zukunft für KMU mit SAP Business One

Daniel Wallner

Technische Hochschule Mittelhessen

Fachbereich MND
Wilhelm-Leuschner-Str. 13
61169 Friedberg
E-Mail:
daniel.wallner@mnd.thm.de

Prof. Dr. Harald Ritz

Technische Hochschule Mittelhessen

Fachbereich MNI
Wiesenstraße 14
35390 Gießen
E-Mail:
harald.ritz@mni.thm.de

Benny Brand

ANG Deutschland GmbH

Geschäftsführung (CEO)
Waldstraße 31
82110 Germering
E-Mail:
benny.brand@an-group.one

Kategorie

Masterarbeit

Schlüsselwörter

SAP Business One, SAP Business Technology Platform, Digitalisierung, KMU, SAP Document Information Extraction, SAP Analytics Cloud, SAP Sustainability Footprint Management, Integration, Side-by-Side-Erweiterungen

Einführung

Die Unternehmenslandschaft in Deutschland ist geprägt von kleinen und mittleren Unternehmen (KMU). Diese haben einen großen wirtschaftlichen und sozialen Einfluss und machen mit rund 3,4 Millionen Unternehmen einen Anteil von 99,2% der Unternehmen in der Privatwirtschaft aus (Stand: 2024). Die KMU stehen vor erheblichen Herausforderungen bei der Digitalisierung ihrer Geschäftsprozesse und müssen ihre Arbeitabläufe optimieren und steigende Anforderungen an die Nachhaltigkei, die Datenanalyse und die Integration von modernen Lösungen erfüllen. Diesen Unternehmen fehlen oft die finanziellen und personellen Ressourcen, um bei der digitalen Transformation mit Großunternehmen mithalten zu können. So beschäftigt aktuell nur knapp jedes fünfte KMU in Deutschland IT-Fachkräfte.

Die SAP Business Technology Platform (BTP) ist eine für SAP-Anwendungen optimierte Cloud-Plattform, die Funktionen wie Datenmanagement, Datenanalysen, künstliche Intelligenz, Anwendungsentwicklung und Automatisierung in einer einheitlichen Umgebung vereint. In Verbindung mit dem ERP-System SAP Business One eröffnen sich speziell für KMU zahlreiche Einsatzmöglichkeiten, um die Unternehmensprozesse zu digitalisieren, die entstehenden Daten effizient zu nutzen und innovative Technologien wie die künstliche Intelligenz einzusetzen. Durch die nahtlose Integration der IT-Landschaften und die zentrale Verwaltung in der Plattform können Unternehmen personelle Ressourcen einsparen und einen Wettbewerbsvorteil erzielen. Im Rahmen dieser wissenschaftlichen Arbeit soll untersucht werden, ob die SAP BTP für KMU mit dem ERP-System SAP Business One im Einsatz, eine sinnvolle Möglichkeit bietet die digitale Transformation voranzutreiben.

Im theoretischen Teil wird der Begriff der kleinen und mittleren Unternehmen erläutert sowie der aktuelle Stand der Digitalisierung in Deutschland untersucht. Des Weiteren werden die technischen Besonderheiten der SAP Business Technology Platform aufzeigt. In diesem Kapitel werden die fünf Kernbereiche der BTP vorgestellt. Die Administration der Plattform erfolgt über das BTP-Cockpit, mit dessen Hilfe der Endanweder den Betrieb und die anfallenden Kosten der abonnierten Lösungen und Serviecs überwachen kann. Ein weiterer Schwerpunkt im theoretischen Teil liegt auf dem ERP-System SAP Business One, insbesondere der vorhandenen Schnittstellen und Erweiterungsmöglichkeiten und der webbasierten Anwendung.

Der nächste Teil der Abschlussarbeit widmet sich den praktischen Einsatzmöglichkeiten der SAP BTP bei KMU. Die technische Integration mit SAP Business One wird detailliert untersucht, ergänzt durch spezifische Lösungen wie das SAP Sustainability Footprint Management, die SAP Document Information Extraction und Erweiterungen im Webclient. Die Lösungen werden durch gezielte Praxisbeispiele veranschaulicht und auf die Umsetzbarkeit und den Nutzen für KMU untersucht. Zusätzlich wird der Einsatz der SAP Analytics Cloud als erweiterte Analyselösung betrachtet. Die Durchführung einer Nutzwertanalyse zwischen der SAP Analytics Cloud und den eingebauten analytischen Funktionen von SAP Business One on HANA dient als Entscheidungshilfe bei der Auswahl eines geeigneten Analysewerkzeugs.

Zusammenfassung

Im Laufe der Arbeit wurden konkrete Anwendungsfälle und technische Lösungen vorgestellt und analysiert. Nachfolgend werden die Kernergebnisse der Lösungen kurz beschrieben.

Das SAP Sustainability Footprint Management ermöglicht die akkurate Erfassung und Analyse des ökologischen Fußabdrucks eines Unternehmens. Die Anwendung unterstützt Unternehmen bei der Darstellung der CO_2-Emissionen entlang der gesamten Wertschöpfungskette sowie bei der Einhaltung von gesetzlichen Vorgaben zur Nachhaltigkeitsberichterstattung. Insbesondere für KMU bietet dies eine praktische Möglichkeit zum Erreichen der unternehmerischen Nachhaltigkeitsziele.

Die Document Information Extraction ist ein Cloud-Service, welcher die automatische Extraktion von strukturierten Informationen aus Dokumenten ermöglicht. Die Lösung bietet zahlreiche praktische Einsatzmöglichkeiten für KMU, wie z.B. die automatische Verarbeitung von Eingangsrechnungen. Durch die Steigerung der Effektivität können personelle Ressourcen entlastet und die Kosten gesenkt werden.

Die SAP BTP ermöglicht weiterhin Erweiterungen im Webclient von SAP Business One. Diese unterstützen KMU bei der branchenspezifischen Anpassung von Prozessen innerhalb des ERP-Systems und tragen zur Verbesserung der Benutzerfreundlichkeit der Software bei. Durch den Side-by-Side-Ansatz, bei dem die Erweiterungen außerhalb des Kernsystems bereitgestellt werden, bleibt die Integrität der IT-Systeme erhalten und Wartungs- und Aktualisierungsaufwände werden reduziert.

Die SAP Analytics Cloud bietet KMU erweiterte Analysefunktionen, die über die native Möglichkeiten von SAP Business One on HANA hinausgehen. Die durchgeführte Nutzwertanalyse hat ergeben, dass die SaaS-Lösung besonders für fortschrittliche Unternehmen mit komplexen Analyseanforderungen oder dem Wunsch nach KI-Unterstützung geeignet ist.

Fazit

Diese Masterarbeit zeigt, dass die SAP BTP eine entscheidene Rolle bei der Digitalisierung von KMU spielen kann. Als zentrale Administrationsplattform ermöglicht die BTP die effiziente Verwaltung von wachsenden heterogenen IT-Landschaften. Die Bereitstellung von innovativen Lösungen und Services durch die BTP schafft eine moderene digitale Infrastruktur und erhöht die Wettbewerbsfähigkeit der Unternehmen. Die Plattform bietet Unternehmen den wesentlichen Vorteil, dass ein Großteil der Sicherheitsverantwortung der IT-Systeme durch die SAP übernommen wird. Dies führt zu einer reduzierten Komplexität und erhöhten Flexibilität für die KMU. Durch die Möglichkeit neue Lösungen oder Services jederzeit hinzuzufügen, erlaubt die Plattform eine kontinuierliche Skalierung der IT-Landschaft. Die Verwendung von KI zur Prozessoptimierung wird in den nächsten Jahren ein zentraler Bestandteil werden, um sich von der Konkurrenz auf dem Markt abzuheben. Der Einsatz von KI wird sich auch in der BTP lösungsübergreifend durchsetzen und erhebliche personelle Einsparungen ermöglichen. Allerdings sollte beachtet werden, dass die Kostenstruktur der BTP für viele KMU eine wesentliche Heraufsorderung darstellen kann. Die Lösungen der Plattform sind häufig an die Bedürfnisse großer Unternehmen angepasst und entsprechen nicht immer den finanziellen Möglichkeiten von KMU. Deshalb müssen die konkreten Anforderungen der Unternehmen individuell betrachtet und den Kosten der BTP-Lösungen gegenübergestellt werden.

Zusammenfassend lässt sich festhalten, dass die SAP BTP ein enormes Potenzial zur Unterstützung der Digitalisierung von KMU bietet, jedoch eine genaue Betrachtung der Kosten erforderlich wird, um eine flächendeckene Nutzung zu ermöglichen.

Literatur

Büchel, Jan, Bakalis, Dennis und Scheufen, Marc (2024). Digitalisierung der Wirtschaft in Deutschland. Digitalisierungsindex 2023 - Langfassung der Ergebnisse des Digitalisierungsindex im Projekt „Entwicklung und Messung der Digitalisierung der Wirtschaft am-Standort Deutschland“. Bundesministerium für Wirtschaft und Klimaschutz (BMWK).

Handel, Holger (2021). Unternehmensplanung mit SAP Analytics Cloud. Rheinwerk-Verlag Bonn.

Institute of Finance and Management (IOFM) (o. D.). *Special Report: The True Costs of Paper-Based Invoice Processing and Disbursements*. https://www.concur.com/sites/default/files/special_report_the_true_costs_of_paper-based_invoice_processing_and_disbursements.pdf

Navandar, Pavan (2023). „Securing Your Applications with Role-Based Access Control in SAP BTP Cockpit“. In: Journal of Artificial Intelligence & Cloud Computing, S. 1–2. DOI: 10.47363/jaicc/2023(2)316.

Seubert, Holger (2024). SAP Business Technology Platform. Einsatz, Services, Konzepte. 3. Auflage. Rheinwerk-Verlag,Bonn.

Wolf, Matthias, Rüdele, Kai und Ramsauer, Christian (2024). „Carbon Footprint Management In Austrian SMEs: Strategies, Mitigation Measures, Challenges, And Case Studies“. In: *Proceedings of the Conference on Production Systems and Logistics*. Hannover : publish-Ing., S. 676–686. DOI: 10.15488/17756.