



Analyse und Bewältigung von Herausforderungen im Software-Prototyping: Eine Untersuchung der Schlüsseldimensionen zur Unterstützung der Digitalen Transformation

Seline Löwe  · Marcel Koch · Thomas Schäffer

Eingegangen: 24. August 2024 / Angenommen: 2. Dezember 2024 / Online publiziert: 7. Januar 2025
© The Author(s) 2025

Zusammenfassung Software-Prototyping von Informationssystemen ermöglicht die frühzeitige Entwicklung von Demonstratoren zur schnellen Integration von Nutzerfeedback, das für die digitale Transformation und organisatorische Veränderungen entscheidend ist. Trotz zahlreicher Vorteile treten bei der Anwendung der Methode Herausforderungen auf, welche sich auch auf die Integration und Verwaltung von Enterprise Systems auswirken. Die Identifizierung von diesen Herausforderungen beim Software-Prototyping ist grundlegend, um Misserfolge bei der Entwicklung innovativer Prototypen zu vermeiden. Mit diesem Beitrag werden fünf Schlüsseldimensionen – Technologie, Organisation, Datenmanagement, Entwicklungsprozess und Benutzererfahrung – durch die Analyse von 39 identifizierten Herausforderungen in 13 Kategorien untersucht. Darüber hinaus zeigen die Ergebnisse, dass in dem achsstufigen Software-Prototyping-Prozess die Phase des User Interface Prototyping die häufigsten Probleme aufweist. Ziel ist es, mögliche Probleme zu identifizieren und zu kategorisieren sowie Handlungsempfehlungen daraus abzuleiten, um ein Bewusstsein für solche zu schaffen, damit Strategien zur digitalen Transformation besser umgesetzt werden können.

Schlüsselwörter Erfolgsfaktoren · Herausforderungen · Prototyping Prozess · Software Prototyping

✉ Seline Löwe · Marcel Koch · Thomas Schäffer
Wirtschaftsinformatik, Hochschule für Angewandte Wissenschaften, Bildungscampus Nord,
74076 Heilbronn, Deutschland
E-Mail: seline.loewe@t-online.de

Marcel Koch
E-Mail: marcelkoch@mailbox.org

Thomas Schäffer
E-Mail: thomas.schaeffer@hs-heilbronn.de

Analysis and Management of Challenges in Software Prototyping: An Examination of Key Dimensions to Support Digital Transformation

Abstract Software prototyping in information systems enables the early development of prototypes for the rapid integration of user feedback which is crucial for digital transformation and organisational changes. Despite numerous advantages, issues arise when using the method—also affecting the integration and management of enterprise systems. Identifying issues in software prototyping is important to prevent failures in the development of innovative prototypes. This study examines five key dimensions—technology, organisation, data management, development process and user experience—by analysing 39 identified issues in 13 categories. Furthermore, findings show that in the eight-stage software prototyping process, the user interface prototyping phase has the most frequent issues. This research aims to identify and categorise possible issues as well as to derive recommendations for action to clarify these and create awareness of such for better digital transformation strategies.

Keywords Issues · Prototyping process · Software prototype · Success factors

1 Einführung

Prototyping ist ein grundlegendes Element des Entwicklungsprozesses von Enterprise Systems (ES) (Khatib et al. 2019). Es bezieht sich auf eine vorläufige Version eines verwendeten Softwaresystems, um bestimmte Aspekte des Systems zu analysieren. Prototyping wird als ein Zusammenspiel verschiedener Formen intellektueller Arbeit gesehen, die sowohl Konzeptualisierung als auch Handlung kombinieren, was für die Integration technologischer Innovationen in Geschäftsprozesse entscheidend ist (Khatib et al. 2019).

Im Zeitalter der Digitalisierung verlangt die globalisierte und dynamische Geschäftswelt eine kurze Markteinführungszeit für Software-Produkte (Gupta et al. 2022). Unternehmen sind auf die schnelle Integration von Nutzerfeedback in ihre ES angewiesen, um sich an dynamische Geschäftsumgebungen und Marktkonkurrenten anzupassen und müssen in der Lage sein, ihre Produkte entsprechend zu adaptieren (Pastoors 2018). Ferner erfordert diese Dynamik eine schnelle Vorhersagbarkeit von Anwendungen (Gupta et al. 2022). Der Einsatz von Prototypen in Informationssystemen ermöglicht frühes Feedback über die Eignung von Lösungen, die Identifizierung von Problemen und Änderungswünschen in einem frühen Stadium (Bjarnason et al. 2023). Dennoch gibt es noch Probleme bei der Anwendung der Prototyping-Methode (Christoforakos et al. 2019). In der Literatur wird thematisiert, dass es derzeit zu wenig Forschung im Bereich Software-Prototyping gibt (Lauff et al. 2018; Gupta et al. 2022) und, dass es wichtig wäre, Probleme beim Software-Prototyping systematisch zu erfassen, um gezielte Handlungsempfehlungen zu generieren (Carfagni et al. 2018; Christoforakos et al. 2019).

Vor diesem Hintergrund zielt unser Beitrag darauf ab, mögliche Herausforderungen systematisch zu identifizieren, kategorisieren und praxisnahe Handlungsempfehlungen abzuleiten. Das frühzeitige Erkennen typischer Problemfelder kann Inno-

vationsprozesse der digitalen Transformation gezielter und effizienter gestalten. Die Handlungsempfehlungen unterstützen Enterprise Systems dabei, Herausforderungen proaktiv anzugehen.

Daraus ergeben sich die folgenden zentralen Forschungsfragen:

- *F1: Welche Herausforderungen treten beim Software-Prototyping auf und welche Handlungsempfehlungen können aus diesen abgeleitet werden?*
- *F2: In welchen Phasen und mit welcher Häufigkeit treten diese Herausforderungen während des Prozesses auf?*

Der vorliegende Beitrag definiert zuerst Prototyping als Methode im Kontext von Informationssystemen. Danach wird die verwendete Forschungsmethodik vorgestellt. Anschließend werden die Ergebnisse erläutert, gefolgt von einer Diskussion. Abschließend werden in einer Schlussfolgerung die Ergebnisse zusammengefasst und Implikationen für Praktiker und Forscher aufgezeigt.

2 Theoretischer Hintergrund

2.1 Software Prototyping als Methode der Wirtschaftsinformatik

Prototyping ist eine etablierte Methode der Produktentwicklung und -innovation, welche in der Industrie weithin akzeptiert ist (Christoforakos et al. 2019).

Das Prototyping ist weniger formalisiert als streng strukturierte Entwicklungsmodelle. Es unterstützt das konstruktive Paradigma von Informationssystemen, da es sich während des Entwicklungsprozesses an die Anforderungen anpasst und die konkrete Erstellung von Artefakten sowie den iterativen Aufbau von Softwarelösungen betont (Wilde und Hess 2007). Diese Flexibilität ist entscheidend für die erfolgreiche Implementierung und Integration von ES, welche häufig eine kontinuierliche Anpassung und iterative Entwicklung erfordern (Leyh und Wendt 2018).

Software-Prototyping ist ein kreatives Verfahren zur Erforschung von Problemen und Lösungen durch die Verwendung von Prototypen. Der Zweck besteht darin, Benutzerprobleme zu identifizieren und Lösungsideen vor der endgültigen Produktentwicklung zu bewerten (Bjarnason et al. 2023).

Duc und Abrahamsson (2016) unterscheiden zwischen drei Arten von Software-Prototypen: Rapide Prototypen erfordern einen minimalen Aufwand und werden hauptsächlich zur Illustration von Ideen verwendet. Evolutionäre Prototypen bilden eine Basisfunktionalität für die weitere Entwicklung des Produkts. Inkrementelle Prototypen beinhalten den schrittweisen Aufbau und die Integration mehrerer Teilprototypen, um das Produkt in überschaubaren Schritten zu entwickeln (Duc und Abrahamsson 2016).

2.2 Software Prototyping Prozess

In diesem Beitrag werden schnelle und inkrementelle Software-Prototypen untersucht und die Anwendung des Prototyping-orientierten Software-Lebenszyklus von Bischofberger und Pomberger (1992) vorgeschlagen.

Dieser Ansatz betont die iterative Entwicklung mit weniger strengen Grenzen zwischen Phasen wie Problemanalyse, Spezifikation, Design, Implementierung und Test. Die Anfangsphase umfasst die Erstellung eines Softwareprototyps auf der Grundlage vorheriger Aktivitäten, wobei der frühen Implementierung von Prototypen Vorrang eingeräumt wird, um iterative Verfeinerungen zu ermöglichen und das Feedback der Benutzer einzubeziehen (Bischofberger und Pomberger 1992).

2.3 Software Prototyping im Bereich Enterprise Systems

Die Veränderungen von Unternehmen aufgrund der fortschreitenden Digitalisierung haben tiefgreifende Auswirkungen auf die Systemlandschaft, von der Implementierung von ERP-Systemen (Enterprise Resource Planning) zur Optimierung von Geschäftsabläufen bis hin zur Integration moderner Technologien. Um den gestiegenen Anforderungen an flexible und effiziente Unternehmenssoftware, insbesondere ERP-Systeme, gerecht zu werden, müssen Unternehmen ihre Informations- und Kommunikationstechnologien anpassen, um im digitalen Zeitalter wettbewerbsfähig zu bleiben (Leyh und Wendt 2018).

Die Prototyping Methode spielt eine entscheidende Rolle im Bereich Enterprise Systems, da es Unternehmen ermöglicht, neue IT-Modelle schnell zu testen und zu verfeinern (Levchenko und Taratukhin 2022). Die Methode bietet Vorteile, wie beispielsweise eine verbesserte Benutzerfreundlichkeit der Anwendung und geringere Entwicklungskosten. Es ermöglicht die Erstellung von Funktionsprototypen und ist für Business-Analysten und UX-Designer vereinfacht zugänglich (Fuhr et al. 2018). Dieser iterative Ansatz ermöglicht es Unternehmen, ihre Enterprise Systems effektiver an spezifische Bedürfnisse anzupassen und Herausforderungen zu bewältigen. Durch den Einsatz von Prototyping können Unternehmen die Agilität ihrer Enterprise Systems erhöhen und Innovationen fördern (Levchenko und Taratukhin 2022).

3 Methodisches Vorgehen

Im Folgenden wird das methodische Vorgehen erläutert, um die Forschungsfragen zu beantworten. Abb. 1 zeigt das methodische Vorgehen.

Die Literatursuche beschränkte die Auswahl der Datenbanken auf internationale Konferenzreihen im Bereich der Informationssysteme, darunter ICIS, ECIS und AMCIS. In der SpringerLink-Datenbank wurde der Filter „Information Systems Applications“ implementiert, um eine präzise Auswahl von Literatur zu gewährleisten, die sich ausschließlich mit Themen im Kontext der Wirtschaftsinformatik beschäftigt. Dies wird durch die Definition von Strangmeier (2008) unterstützt, der Informationssysteme und ihre Anwendungen als zentrale Bestandteile von Informationssystemen beschreibt.

Als zentrales Element der vorliegenden Literaturanalyse wurde der Suchbegriff „Prototyping“ AND („Information Systems“ OR „Challenges“ OR „Method“ OR „Application“) verwendet, der entweder im Titel oder in der Zusammenfassung der wissenschaftlichen Arbeiten vorkommt. Der Suchbegriff wurde gewählt, um gezielt

wissenschaftliche Arbeiten zu identifizieren, die Prototyping in Verbindung mit den Bereichen Informationssysteme, Herausforderungen, Methoden oder Anwendungen thematisieren. Zusätzlich wurde ein Zeitfilter angewendet, um den Fokus auf State-of-the-Art-Publikationen zu legen, weshalb nur Arbeiten berücksichtigt wurden, deren Veröffentlichung zwischen 2018 und 2023 liegt.

Die Literatursuche ergab insgesamt 225 Referenzen. Zunächst wurde der Titel analysiert und die Zugänglichkeit geprüft, sodass insgesamt 58 Referenzen übrigblieben. In einem nächsten Schritt wurden Duplikate entfernt, wodurch sich die Zahl auf 39 reduzierte. Die verbleibenden Referenzen wurden dann auf ihre wissenschaftliche Eignung hin überprüft. Die Entscheidung über die wissenschaftliche Eignung des Inhalts wurde durch die Analyse von Schlüsselwörtern und der Zusammenfassung getroffen. Darüber hinaus wurden die Forschungsarbeiten untersucht, ob sie sich mit Fragen des Software-Prototyping-Prozesses befassen. Wenn dies nicht der Fall war, wurden sie verworfen. Durch eine Vorwärts- und Rückwärtssuche nach Watson und Webster (2002) wurden weitere fünf Referenzen ermittelt. Letztendlich führte dieser Prozess zu einer Gesamtzahl von 17 relevanten Publikationen.

Die Analyse der relevanten Publikationen erfolgte auf Basis der qualitativen Inhaltsanalyse nach Mayring (2010). Dazu wurden alle als relevant identifizierten Publikationen zunächst auf ihre Artefakte hin untersucht. Anhand dieser Inhalte wurde festgestellt, ob es sich um Software-Prototypen handelt. Die Softwareprototypen wurden dann in die drei Typen nach Duc und Abrahamsson (2016) kategorisiert und den jeweiligen Phasen des Prototyping-orientierten Softwarelebenszyklus nach Bischofberger und Pomberger (1992) zugeordnet.

Im nächsten Schritt analysierten wir die Publikationen auf Probleme im Software-Prototyping-Prozess. Für eine detailliertere Analyse der einzelnen Textsegmente verwendeten wir das Verfahren der thematischen Synthese nach Cruzes und Dybá (2011), das zu insgesamt 46 Textsegmenten führte. Anschließend haben wir diese Textsegmente in 39 Themen gefiltert und diese in dreizehn Variablen unterteilt, die wiederum fünf Dimensionen zugeordnet wurden. Die Variablen und Dimensionen haben wir mit der induktiven Kategorienentwicklungsmethode nach Mayring (2010) bestimmt und das Vier-Augen-Prinzip nach Peffers et al. (2012) angewen-

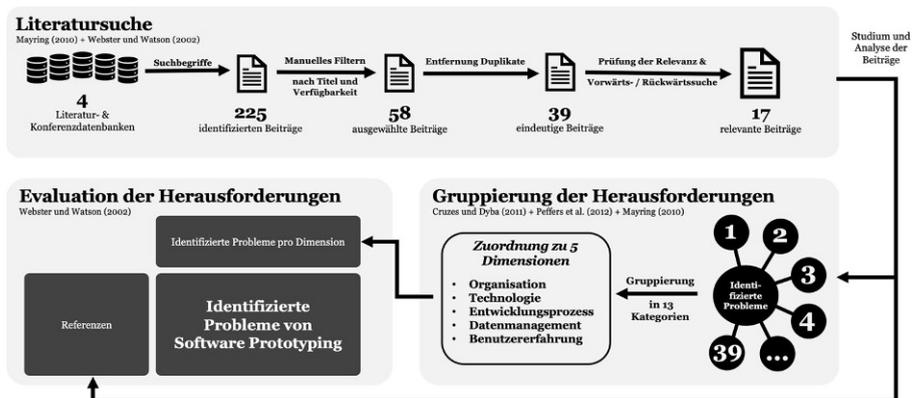


Abb. 1 Überblick über das methodische Vorgehen

det. Schließlich erstellten wir eine Konzeptmatrix nach Webster und Watson (2002), um die Häufigkeit von Problemen in den Referenzen zu bewerten, sowie ein geclustertes Balkendiagramm, um das Auftreten und die Häufigkeit von Problemen im Prototyping-orientierten Software-Lebenszyklus nach Bischofberger und Pomberger (1992) zu identifizieren. Die Referenzen dieses Beitrags wurden entsprechend in die acht Phasen des Prototyping-orientierten Software-Lebenszyklus eingeteilt: Anforderungsanalyse, Anforderungsdefinition, User Interface Prototyping, Architektur- und Komponentenentwurf, Architektur- und Komponentenprototyping, Implementierung, Systemtest, Betrieb und Wartung.

Die in dieser Literaturanalyse verwendeten Referenzen wurden in eine oder zwei Phasen eingeordnet, wobei keine der Quellen speziell den ersten beiden Phasen zugeordnet wurde. Alle Referenzen beziehen sich auf einen aktiven Software-Prototyping-Prozess, der bereits stattfindet. Es muss also vorher eine Anforderungsanalyse und -definition stattgefunden haben. Die vorliegende Arbeit deckt somit alle Phasen des Prototyping-orientierten Software Life Cycle nach Bischofberger und Pomberger (1992) ab. Die Einordnung der Literatur ist kein Bestandteil dieser Arbeit, kann jedoch bei den Autor*innen angefragt werden.

4 Ergebnisse: Identifizierte Herausforderungen

Im folgenden Abschnitt werden die anhand der Literaturanalyse identifizierten Dimensionen Organisation (Orga), Technologie (Tech), Entwicklungsprozess (Dev-Pro), Datenmanagement (DaMa) und Benutzererfahrung (UX) in der Reihenfolge ihrer relativen Häufigkeit, von häufig bis selten, dargestellt. Die relative Häufigkeit ergibt sich aus der Gesamtzahl der identifizierten Herausforderungen (39) als Nenner und der Häufigkeit des Auftretens der jeweiligen Herausforderung innerhalb der einzelnen Dimensionen als Zähler. Die 39 identifizierten Herausforderungen werden detailliert innerhalb der 13 Kategorien erläutert. Die Kategorien repräsentieren die übergeordneten Herausforderungen und werden jeweils einer der fünf Dimensionen zugeordnet. Die Konzeptmatrix nach Webster und Watson (2002) wird in Tab. 1 dargestellt und bietet einen Überblick über die identifizierten Herausforderungen im Software-Prototyping sowie deren relativer Häufigkeit.

4.1 Organisation

Die Organisation des Prototyping-Prozesses ist entscheidend für den Erfolg eines Prototyps. Bei Enterprise Systems wird dies durch hohe Anforderungen und Komplexität noch wichtiger. Folgende organisatorische Aspekte wurden ermittelt.

4.1.1 Mangelhaftes Training

Eine zentrale Herausforderung im Prototyping-Prozess ist der Mangel an Schulung und Vorbereitung, insbesondere für unerfahrene Teams (da Silva Melo et al. 2022).

Ein Mangel an Schulung und Unterstützung kann die Einführung neuer Technologien behindern (Heil et al. 2018). Vor allem für kleinere Unternehmen können

Tab. 1 Überblick über die identifizierten Herausforderungen im Software Prototyping

Referenzen	Orga			Tech			DevPro		DaMa		UX		
	Mangelhaftes Training	Mangel an Fachwissen	Mangelhafte Guidelines	Technische Abhängigkeit	Umfassende Systemintegration	Systemkomplexität	Unzureichendes Zeitmanagement	Unvollständige Anforderungen	Unzureichende Datenqualität	Komplexe Datenintegration	Unvollständige Datensicherheit	Unbefriedigende Benutzerfreundlichkeit	Fehlende Akzeptanz
Ahmed und Demirel 2021	X	X	X	-	-	-	-	-	-	-	-	-	-
Bertola 2021	-	-	-	-	-	-	-	X	X	-	-	-	-
Croft et al. 2019	-	-	-	-	X	-	-	-	-	X	-	-	-
Da Silva Melo et al. 2022	X	-	-	-	-	-	-	-	-	-	-	-	-
Feldmann und Teutenberg 2019	X	-	X	X	-	-	X	-	-	-	-	-	-
Gerassimov et al. 2020	-	-	-	-	-	-	X	X	-	-	-	-	-

Tab. 1 (Fortsetzung)

Referenzen	Orga		Tech		DevPro		DaMa		UX				
	Mangelhaftes Training	Mangel an Fachwissen	Mangelhafte Guidelines	Technische Abhängigkeit	Umfassende Systemintegration	Systemkomplexität	Unzureichendes Zeitmanagement	Unvollständige Anforderungen	Unzureichende Datenqualität	Komplexe Datenintegration	Unvollständige Datensicherheit	Unbefriedigende Benutzerfreundlichkeit	Fehlende Akzeptanz
Heil et al. 2018	X	X	-	-	-	-	-	-	-	-	-	-	-
Hicking 2020	-	-	-	-	-	-	-	-	-	X	X	-	-
Klippenstein et al. 2023	-	X	-	-	-	-	-	X	-	-	-	-	-
Köbel et al. 2023	-	X	-	-	X	-	X	-	-	-	-	-	-
Piazas et al. 2020	-	-	-	-	X	-	-	-	-	-	-	-	-
Qiu et al. 2018	-	-	X	-	-	-	X	-	-	-	-	X	-

Tab. 1 (Fortsetzung)

Referenzen	Orga		Tech		DevPro		DaMa		UX				
	Mangelhaftes Training	Mangel an Fachwissen	Mangelhafte Guidelines	Technische Abhängigkeit	Umfassende Systemintegration	Systemkomplexität	Unzureichendes Zeitmanagement	Unvollständige Anforderungen	Unzureichende Datenqualität	Komplexe Datenintegration	Unvollständige Datensicherheit	Unbefriedigende Benutzerfreundlichkeit	Fehlende Akzeptanz
Rupprecht et al. 2018	-	-	-	-	X	X	-	-	-	-	-	-	-
Shakeel et al. 2022	-	-	-	-	-	-	X	-	-	-	-	-	-
Vogel et al. 2021	-	-	-	X	-	-	-	-	-	-	-	X	X
Wu et al. 2018	X	-	-	X	-	-	-	-	X	-	-	-	-
Zhang et al. 2022	-	-	-	X	-	X	-	-	-	-	-	-	-
Relative Häufigkeit	31 %		28 %			18 %			15 %			8 %	

Fragen wie fehlendes Fachwissen problematisch sein, da ihre Fähigkeit, Prototypen effizient zu erstellen, beeinträchtigt werden kann (Feldmann und Teuteberg 2019). Darüber hinaus bedeutet eine unzureichende Ausbildung von Designern und Ingenieuren, dass sie ihre Kreativität und Intuition nutzen, um Prototypen zu erstellen, anstatt eine objektive Strategie zu verfolgen, welche durch Schulungen vermittelt wird. Dies kann insbesondere bei Enterprise Systems für große Unternehmen den entscheidenden Vorteil in der Effizienz und Entwicklungszeit verhindern (Ahmed und Demirel 2021). Die aktive Schulung der Anforderungsinduktion ist ebenfalls eine Herausforderung, da sich die Anforderungen häufig ändern oder unvollständig definiert sind (da Silva Melo et al. 2022). Ein Mangel an Training kann auch auf der technischen Seite auftreten, da beispielsweise das Training neuronaler Netze zeitaufwändig ist und die Antwortqualität stark beeinflusst (Wu et al. 2018).

4.1.2 *Mangel an Fachwissen*

Der Einsatz spezifischer Kodierungsmethoden kann im Prototyping zwar effektiv sein, jedoch birgt er die Gefahr, eine große Anzahl an Merkmalen zu erzeugen, was ohne ausreichendes Fachwissen die Verallgemeinerung erschweren kann (Klippenstein et al. 2023). Eine weitere Herausforderung ist die effektive Übertragung von Designwissen auf Prototypen, insbesondere wenn Entscheidungen auf subjektiven Präferenzen beruhen (Kölbel et al. 2023). Des Weiteren können Bedenken hinsichtlich der Erwünschtheit auf mangelndem Wissen über aktuelle Webtechnologien und deren Fähigkeiten beruhen, was zu Unsicherheit und potenziell suboptimalen Designentscheidungen führt (Heil et al. 2018). Unerfahrene Mitarbeiter nutzen oft ihre Kreativität und Intuition, um Prototypen zu erstellen. Insbesondere in Enterprise Systems können Fehler aufgrund von geringerer Softwarequalität direkte Auswirkungen auf Geschäftsprozesse und damit auf das Geschäftsergebnis haben (Ahmed und Demirel 2021).

4.1.3 *Mangelhafte Guidelines*

Eine wichtige Herausforderung auf organisatorischer Ebene ist die Validierung von Ideen (Feldmann und Teuteberg 2019). Das Fehlen von Fertigungsrichtlinien ist ebenso zu beachten. Oft bieten Prototyping-Frameworks keine spezifischen Richtlinien für die Entwicklung (Ahmed und Demirel 2021). Zudem kann ein Mangel an Leitlinien für den gezielten Einsatz von Humanressourcen zu unausgewogenen Ergebnissen und eingeschränkter Validität im Entwicklungsprozess führen (Qiu et al. 2018).

4.2 **Technologie**

Für die Entwicklung eines Software-Prototyps sind auch bestimmte technologische Voraussetzungen erforderlich. Im Hinblick auf die technologische Dimension wurden die folgenden Punkte ermittelt.

4.2.1 Technische Abhängigkeit

Bereits scheinbar kleine technische Details können im Prototyping-Prozess erhebliche Auswirkungen haben (Vogel et al. 2021). Prototypen für Enterprise Systems müssen sich dynamisch zum Unternehmenswachstum und der damit verbundenen steigenden Belastung anpassen. Dies erschwert nicht nur die Entwicklung, sondern kann auch zu unerwarteten Problemen führen, wenn z. B. die Dateigrößen schnell ansteigen (Zhang et al. 2022; Wu et al. 2018). Darüber hinaus können die Konfiguration einer großen Anzahl von Datenzentren und die Auswahl der technologischen Komponenten während des Prototyping-Prozesses zu erheblichen Schwierigkeiten führen (Zhang et al. 2022; Hicking 2020). Die Ungewissheit über sich verändernde Technologien führt zu Zurückhaltung von größeren Investitionen bei Unternehmen und erschwert damit auch die Vorhersage von Entwicklungsrichtungen sowie die Festlegung auf bestimmte Technologien (Feldmann und Teuteberg 2019).

4.2.2 Umfassende Systemintegration

Auch die Systemintegration stellt Praktiker vor eine Vielzahl von Problemen. Die Integration bestehender oder veralteter Informationssysteme in ein fortschrittliches User-Interface-Prototyping-System erfordert in der Regel einen erheblichen Arbeitsaufwand, um die Daten aus dem System selbst zu übernehmen. In Großunternehmen verursacht diese Migration in ein neues System oft die größten Kosten im Projekt (Croft et al. 2019; Plazas et al. 2020). Reputationsmechanismen ermöglichen Nutzern die Kontrolle über ihre Daten zu behalten und einen sicheren Zugang zu Diensten zu gewährleisten, jedoch kann die Integration eine Herausforderung darstellen (Kölbel et al. 2023). Außerdem kann die Verwendung unterschiedlicher Softwaretools zu Inkonsistenzen in der Semantik des gesamten Anwendungsdesigns führen (Rupprecht et al. 2018).

4.2.3 Systemkomplexität

Die Systemkomplexität stellt in der Softwareentwicklung eine zentrale Herausforderung dar, da sie Leistungseinbußen verursachen kann und die Anpassung an Unternehmensanforderungen erschweren kann. Zudem kann die tatsächliche Leistung hinter den theoretischen Erwartungen zurückbleiben (Zhang et al. 2022). Ein weiteres Problem ist, dass die Verwendung eines einzigen neueren Softwaretools zwar den Vorteil bietet, dass die Nutzenden von einer einheitlichen Terminologie und Benutzeroberfläche profitieren, diese sich jedoch erst mit einem neuen Designtool vertraut machen und ihre vertrauten Tools beiseitelegen müssen (Rupprecht et al. 2018).

4.3 Entwicklungsprozess

Die Dimension des Entwicklungsprozesses verdeutlicht die verschiedenen Herausforderungen, welche bei der Entwicklung und Umsetzung von Prototypen auftreten können.

4.3.1 Unzureichendes Zeitmanagement

Ein unzureichendes Zeitmanagement im Entwicklungsprozess ist eine häufige Herausforderung, welche die Effizienz und den Erfolg von Prototyping-Projekten beeinträchtigen kann. Besonders in Großprojekten kann ein kontinuierlicher Überprüfungsprozess das Entwicklungstempo für das Design und die Entwicklung von Prototypen beschleunigen (Shakeel et al. 2022; Kölbel et al. 2023). Allerdings ist jeder Schritt ein ressourcenintensiver Prozess, der eine längere Entwicklungszeit erfordert (Feldmann und Teuteberg 2019). In der Praxis besteht ein hoher Bedarf, die Entwicklungszeit zu verkürzen und die Markteinführung zu beschleunigen, um so der Konkurrenz zuvorzukommen (Gerasimov et al. 2020). Ein weiterer Aspekt ist die Notwendigkeit, Usability-Probleme frühzeitig zu erkennen, um die Entwicklungszeit zu verkürzen (Qiu et al. 2018).

4.3.2 Unvollständige Anforderungen

Je nach Zielgruppe von Enterprise Systems sind die Anforderungen vielseitig und verschiedenste Stakeholder müssen in die Sammlung und Qualifizierung der Anforderungen miteinbezogen werden. Unvollständige Anforderungen erschweren den Entwicklungsprozess, da vorläufige Oberflächendesigns oft schnell veralten und später Anpassungen am Datenmodell zur Verbesserung der Nutzererfahrung nötig machen (Gerasimov et al. 2020). Unvollständige Anforderungen verringern auch die Effizienz des Entwicklungsprozesses und führen oft zu einem Endergebnis, welches die Erwartungen der Endnutzer nicht erfüllt (Bertola 2021). Wenn nicht alle Anforderungen klar definiert sind oder nicht verstanden werden, können Inkompatibilitäten oder Fehlfunktionen auftreten, die später im Entwicklungsprozess behoben werden müssen und vermeidbare Kosten für die Unternehmen verursachen (Gerasimov et al. 2020).

4.4 Datenmanagement

Im Prototyping in Enterprise Systems ist ein effektives Datenmanagement entscheidend, um die Qualität, Integration und Sicherheit der Daten sicherzustellen und eine stabile Grundlage für die Systementwicklung zu schaffen.

4.4.1 Unzureichende Datenqualität

Ein mangelndes gemeinsames Verständnis der Bewerter könnte zu unerwünschten Verzerrungen in den Bewertungsdaten und somit zu einer geringeren Datenqualität führen (Klippenstein et al. 2023). Ein weiteres Problem für die Datenqualität ist die inkonsistente Einrichtung von Big-Data-Plattformen, da eine unsaubere Datenintegration die Ergebnisse negativ beeinflussen kann (Wu et al. 2018). Bertola (2021) betont, dass die Integration heterogener Daten in ein einheitliches digitales Modell, etwa bei BIM, eine große Herausforderung für die Datenqualität darstellt, da die Daten diskretisiert werden müssen. Darüber hinaus weisen beispielsweise Crowdsourcing-Plattformen oft eine schwankende Datenqualität auf. Dies hebt die

Bedeutung von Qualitätssicherung und Datenbereinigung hervor (Klippenstein et al. 2023).

4.4.2 Komplexe Datenintegration

Die Integration komplexer Daten bringt eine Reihe von Herausforderungen mit sich, insbesondere wenn es um die Sicherheit und den Schutz sensibler oder klassifizierter Informationen geht (Croft et al. 2019). Die Gewährleistung der Informationssicherheit ist daher eine zentrale Herausforderung bei der Datenerhebung und -verarbeitung (Hicking 2020). Die Integration von Daten aus verschiedenen Quellen erfordert eine sorgfältige Handhabung, da diese sonst beim Integrationsprozess gefährdet werden können (Hicking 2020).

4.4.3 Unvollständige Datensicherheit

Beim Prototyping liegt der Schwerpunkt häufig auf der schnellen Konzeptentwicklung und der Demonstration der Funktionalität. Ein Mangel an Datensicherheit während des Prototyping-Prozesses kann schwerwiegende Folgen haben (Hicking 2020). Unzureichende Sicherheitsmaßnahmen können zu Datenschutzverletzungen, Datenlecks oder unbefugtem Zugriff führen, was das Vertrauen der Nutzer gefährden und rechtliche Konsequenzen nach sich ziehen kann. Die ab 2018 gültige europäische Datenschutzgrundverordnung bestraft Unternehmen bei Nichtbeachtung mit Bußgelder von bis zu 10 Mio. € oder 2% des Unternehmensjahresumsatzes im vorangegangenen Geschäftsjahr (Artikel 83 Absatz 4 DSGVO).

4.5 Benutzererfahrung

Die Benutzererfahrung ist ein Schlüsselement im Entwicklungsprozess von Software-Prototypen. Im Folgenden werden die zentralen Herausforderungen in der Benutzererfahrung – unbefriedigende Benutzerfreundlichkeit und fehlende Akzeptanz – gezielt beleuchtet, da diese entscheidend für die Zufriedenheit und Effektivität von Software-Prototypen im Entwicklungsprozess sind.

4.5.1 Unbefriedigende Benutzerfreundlichkeit

Herausforderungen bei der Benutzeroberfläche und Aktionen, welche Training erfordern, zeigen die Wichtigkeit gründlicher Usability-Tests und klarer Richtlinien, bevor Prototypen Nutzenden bereitgestellt werden (Vogel et al. 2021). Vor allem in schnelllebigen Entwicklungsprozessen, wie der Erstellung von Web- und Mobilanwendungen, kann eine Expertenevaluierung von entscheidender Bedeutung sein, um potenzielle Probleme frühzeitig zu erkennen und zu beheben, bevor der Prototyp live geht (Qiu et al. 2018).

4.5.2 Fehlende Akzeptanz

Die Technologieakzeptanz ist entscheidend für die Benutzererfahrung und stellt in Großunternehmen mit vielseitiger Belegschaft eine komplexe Herausforderung dar. Neue Ansätze wie Design Thinking in virtueller Realität stoßen aufgrund invasiver Hardware besonders bei älteren oder weniger digital versierten Nutzenden auf Akzeptanzprobleme (Vogel et al. 2021). Daher ist es von entscheidender Bedeutung, die Technologieakzeptanz genauer zu untersuchen, insbesondere wenn der Prototyp in einer realen Fallstudie eingeführt wird, zum Beispiel in Zusammenarbeit mit einem Design Thinking Facilitator (Vogel et al. 2021).

4.6 Häufigkeitsverteilung der Herausforderungen innerhalb des Software-Prototyping Prozesses

Die Analyse der Häufigkeitsverteilung von Herausforderungen im achtstufigen Software-Prototyping-Prozess nach Bischofberger und Pomberger (1992) liefert Erkenntnisse über das Auftreten von Herausforderungen im Entwicklungsprozess. Die dritte Phase ist besonders durch Herausforderungen gekennzeichnet. Mindestens zwei pro Dimension treten in dieser Phase auf. Dies deutet darauf hin, dass Phase drei eine kritische Phase im Entwicklungsprozess ist, in der eine Vielzahl von Aspekten und Anforderungen berücksichtigt werden muss. Nach der dritten Phase nimmt die Häufigkeit der Herausforderungen in den nachfolgenden Phasen deutlich ab. Dies deutet darauf hin, dass die Komplexität und die Anforderungen der verschiedenen Dimensionen des Prototyping-Prozesses in den späteren Phasen variieren können. Eine weitere interessante Beobachtung ist, dass in der Systemtestphase erneut Probleme in allen Phasen auftreten, mit Ausnahme der Datenverwaltung. Schlussfolgernd, ist die Testphase eine entscheidende Phase, in der verschiedene Aspekte des Prototyps erneut überprüft werden müssen. Das folgende Balkendiagramme in Abb. 2 zeigt die Verteilung der identifizierten Herausforderungen innerhalb des Prototyping-orientierten Software-Lebenszyklus nach Bischofberger und Pomberger (1992).

5 Diskussion und Handlungsempfehlungen

Wie eingangs erläutert, besteht ein häufig angeführtes Problem in der mangelnden Forschung zu Herausforderungen beim Software-Prototyping und der daraus resultierenden Schwierigkeit, konkrete Handlungsempfehlungen abzuleiten. Unsere Analyse zeigt jedoch, dass entgegen dieser Annahme eine Vielzahl an wissenschaftlicher Literatur existiert, welche insgesamt 39 spezifische Herausforderungen identifizieren. Allerdings fehlt es weitgehend an Literatur, die klare Empfehlungen zur praktischen Umsetzung dieser Erkenntnisse bietet. Zur Schließung dieser Lücke, haben wir aus den identifizierten Herausforderungen innerhalb der fünf Dimensionen gezielte Handlungsempfehlungen für ES abgeleitet. Im Folgenden werden diese Herausforderungen und ihr Auftreten in den einzelnen Phasen analysiert und praxisnahe Handlungsempfehlungen abgeleitet. Die häufigsten Herausforderungen

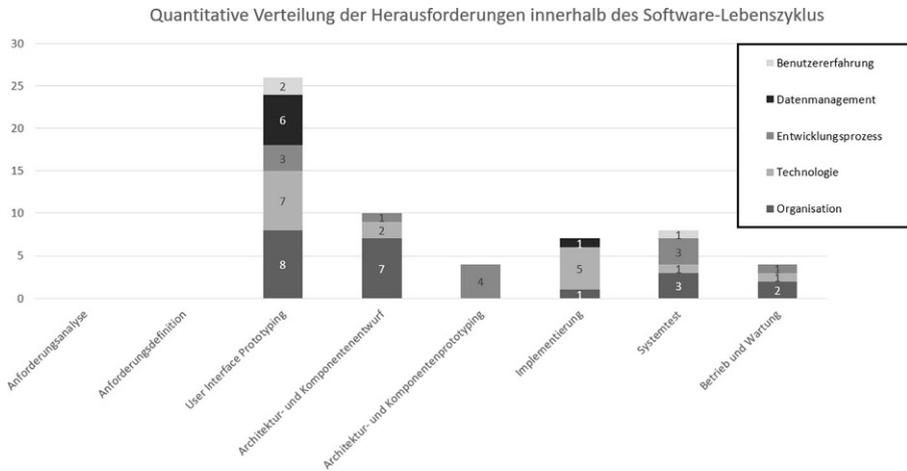


Abb. 2 Häufigkeitsverteilung der Herausforderungen innerhalb des Prototyping-orientierten Software-Lebenszyklus nach Bischofberger und Pomberger (1992)

betreffen die organisatorische Dimension, insbesondere den Mangel an spezifischem Training, Fachwissen und klaren Richtlinien. Diese Defizite führen oft zu ineffizienten Prototyping-Prozessen und erschweren es Teams, eine konsistente und zielgerichtete Vorgehensweise zu entwickeln. Im Gegensatz dazu treten in der Dimension der Benutzererfahrung vergleichsweise weniger Herausforderungen auf, was darauf hindeutet, dass viele Unternehmen bereits grundlegende Maßnahmen zur Verbesserung der UX implementiert haben. Folgende Handlungsempfehlungen können aus den Inhalten der 39 Herausforderungen innerhalb der fünf Dimensionen abgeleitet werden:

Organisation Führen Sie regelmäßige Schulungen sowie standardisierte Prototyping-Richtlinien ein. Diese Maßnahmen stärken das Wissen und die Kompetenzen der Teams in komplexen Unternehmensstrukturen.

Technologie Etablieren Sie technische Standards sowie einen modularen und skalierbaren Entwicklungsansatz. Integrieren Sie außerdem eine kontinuierliche Technologiebetrachtung, um frühzeitige Anpassungen vornehmen zu können. Diese Strategien unterstützen vor allem die Benutzeroberflächenphase und machen Enterprise Systems widerstandsfähiger gegenüber sich verändernden Anforderungen.

Entwicklungsprozess Implementieren Sie agile Methoden, wie beispielsweise Scrum oder Kanban, um iterative Prozesse in Enterprise Systems zu unterstützen. Flexible Ansätze können Teams besser auf unvollständige Anforderungen sowie Zeitengpässe vorbereiten, insbesondere in den systemkritischen Phasen wie Systemtests und Benutzeroberflächenentwicklung.

Datenmanagement Entwickeln Sie strikte Standards für die Datenintegration und implementieren Sie diverse Sicherheitsmaßnahmen. Enterprise Systems verwalten

oftmals sensible und umfangreiche Unternehmensdaten, weshalb ein datenorientiertes Sicherheitssystem, welches den gesetzlichen Standards entspricht, unverzichtbar ist. Dies ist besonders in den datenintensiven Phasen der Systemintegration und des Systemtests entscheidend.

Benutzererfahrung Führen Sie regelmäßige Usabilitytests und iterative Anpassungen durch. Ein frühzeitiges Feedback sowie ein nutzerzentriertes Design sind bei Enterprise Systems wichtig, um UX-Probleme bereits in der Prototyping-Phase zu identifizieren und zu beheben.

Zudem zeigte sich, dass es bisher kaum Literatur gibt, die eine Einordnung der Herausforderungen entlang des Entwicklungsprozesses innerhalb der einzelnen Phasen vornimmt. Daher ergänzen wir die Forschung durch eine systematische Einordnung dieser Herausforderungen. Die Analyse der Verteilung der Herausforderungen zeigt, dass die Prototyping-Phase der Benutzeroberfläche die meisten Probleme aufweist, wobei alle Dimensionen betroffen sind. Insgesamt wurden in dieser Phase 26 spezifische Herausforderungen festgestellt. Im Gegensatz dazu weist die Phase des Architektur- und Komponentenentwurfs nur Herausforderungen in drei Dimensionen auf. Interessant ist jedoch, dass in der Systemtestphase alle Dimensionen vertreten sind, mit Ausnahme des Datenmanagement. Dies könnte darauf hindeuten, dass Herausforderungen mit dem Datenmanagement in früheren Phasen des Prozesses gelöst wurden.

Unsere Studie hat jedoch auch Limitationen. Die subjektive Herleitung und Interpretation durch uns Autoren kann zu Verzerrungen führen. Außerdem können kulturelle oder regionale Unterschiede in der Prototyping-Praxis zu zusätzlichen Variationen führen, welche in unserer Studie nicht berücksichtigt wurden. Bei der Literaturrecherche wurde auch eine Vielzahl von Arbeiten identifiziert, die andere Prototypentypen als die drei untersuchten beschreiben, und der Prototyping-Prozess von Bischofberger und Pomberger (1992) ist nur eine Version von zahlreichen anderen.

6 Zusammenfassung

In unserem Beitrag zeigen wir 39 Herausforderungen in 13 Kategorien und fünf Dimensionen des Software-Prototyping-Prozesses auf. Es wird betont, wie wichtig es ist, Prototyping-Herausforderungen in den fünf Dimensionen – Organisation, Technologie, Entwicklungsprozess, Datenmanagement und Benutzererfahrung – mit unseren praxisnahen Handlungsempfehlungen anzugehen, um eine erfolgreiche Implementierung und Integration von Enterprise Systems zu gewährleisten, die digitale Transformation zu fördern und die organisatorischen Abläufe zu optimieren. Zudem wird hervorgehoben, dass die Prototyping-Phase der Benutzeroberfläche des achtstufigen Prototyping-orientierten Software-Lebenszyklus von Bischofberger und Pomberger (1992) die meisten Probleme aufweist, wobei alle Dimensionen vertreten sind. Es fällt auf, dass sich die Probleme danach tendenziell auflösen und in den einzelnen Phasen des Prozesses gelegentlich wieder auftauchen.

Dennoch gibt es Einschränkungen, wie mögliche Verzerrungen in der Interpretation und eine begrenzte Abdeckung verschiedener Prototyping Praktiken. Daraus leiten wir Implikationen für Praktiker und Forscher im Bereich Enterprise Systems ab.

6.1 Implikationen für die Wissenschaft

Die im Software-Prototyping-Prozess ermittelten Herausforderungen haben auch im Kontext von Enterprise Systems erhebliche Forschungsrelevanz. In der Wissenschaft sollten insbesondere technologische Abhängigkeiten wie Schnittstellenkompatibilität, Datenmigration, und die Integration von Legacy-Systemen berücksichtigt werden, da diese entscheidend für die erfolgreiche Implementierung neuer Technologien in komplexe Enterprise Systems sind. Im Bereich des Datenmanagements könnten innovative Ansätze wie die Implementierung von Blockchain-Technologie zur Verbesserung der Datensicherheit oder der Einsatz von API-Gateways zur Optimierung der Datenintegration erforscht werden. Darüber hinaus ist die Untersuchung der Benutzererfahrung von Prototypen in Enterprise Systems entscheidend, um spezifische Bedürfnisse der Anwender frühzeitig besser zu verstehen. Besonders zu beachten sind dabei die Anforderungen an intuitive Benutzeroberflächen, nahtlose Integration in bestehende Workflows sowie die Möglichkeit zur Personalisierung der Benutzererfahrung. Ein Vergleich des Lebenszyklus von Bischofberger und Pomberger (1992) mit modernen Ansätzen könnte zudem aufzeigen, wie sich die Herausforderungen im Prototyping über die Zeit verändert haben. Folglich ist die Erforschung der Wechselwirkungen zwischen diesen Herausforderungen wichtig, um ganzheitliche Lösungen zu entwickeln, die die Effizienz und Qualität von Prototypen in Enterprise Umgebungen verbessern.

6.2 Implikationen für die Praxis

Die ermittelten Herausforderungen beim Software-Prototyping haben auch konkrete Auswirkungen für die Praxis, insbesondere für Softwareentwickler, Projektmanager und Entscheidungsträger in Unternehmen. Praktiker sollten sich über spezifische technologische Abhängigkeiten im Klaren sein, welche den Prototyping Prozess beeinflussen können. Besonders zu berücksichtigen sind hierbei die Interoperabilität zwischen verschiedenen Systemen, die Abhängigkeit von Cloud-Diensten und die Integration von Drittanbieter APIs. Eine sorgfältige Bewertung der verfügbaren Technologien in Bezug auf Kosten und Nutzen und eine darauf basierende fundierte Auswahl derjenigen, welche den Anforderungen des Projekts am besten entsprechen, sind entscheidend. Hinsichtlich der Sicherheit sensibler Informationen während des Prototyping Prozesses sollten konkrete Maßnahmen ergriffen werden. Dazu zählen die Verschlüsselung von Daten, die Implementierung von Zugriffssteuerungen sowie regelmäßige Sicherheitsüberprüfungen, um Daten vor Verlust, Diebstahl oder unbefugtem Zugriff zu schützen. Auch die frühzeitige Einbindung eines Datenschutzbeauftragten ist empfehlenswert, um später notwendige Korrekturaufwände aufgrund Nichtbeachtung der DSGVO und Bußgelder zu vermeiden. In Anbetracht der Bedeutung von Wissen und Ressourcen für erfolgreiche Prototyping-Prozesse sollten

Organisationen sicherstellen, dass ihre Teams angemessen geschult sind und über die notwendigen Ressourcen verfügen. In einigen Fällen ist es auch sinnvoll, Fachleute für bestimmte Enterprise Anwendungen in den Prototyping-Prozess einzubeziehen, um sicherzustellen, dass der Prototyp später reibungslos in die bestehende IT-Landschaft integriert werden kann. Die Bedeutung eines effizienten Zeitmanagements im Prototyping-Prozess sollte ebenfalls nicht unterschätzt werden. Praktiker sollten sicherstellen, dass für jeden Schritt des Prozesses ausreichend Zeit eingeplant wird, um Verzögerungen zu vermeiden und die Markteinführung zu beschleunigen. In Enterprise Umgebungen können Verzögerungen im Prototyping aufgrund der Systemabhängigkeit erhebliche Auswirkungen haben. Die Identifizierung von Problemen der Benutzererfahrung sollte eine hohe Priorität haben. Die Nutzer von Enterprise Anwendungen sind oft Fachkräfte, welche auf effiziente und intuitive Benutzeroberflächen angewiesen sind. Probleme in diesem Bereich frühzeitig zu erkennen und zu adressieren, kann maßgeblich dazu beitragen, die Akzeptanz und Zufriedenheit der Endnutzer zu steigern und damit die Konkurrenzfähigkeit zu erhöhen.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access Dieser Artikel wird unter der Creative Commons Namensnennung 4.0 International Lizenz veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden. Die in diesem Artikel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen. Weitere Details zur Lizenz entnehmen Sie bitte der Lizenzinformation auf <http://creativecommons.org/licenses/by/4.0/deed.de>.

Literatur

- Ahmed S, Demirel HO (2021) A Prototyping framework for human-centered product design: preliminary validation study. In: Soares MM, Rosenzweig E, Marcus A (Hrsg) Design, user experience, and usability: UX research and design. HCII 2021. Lecture Notes in Computer Science, Bd. 12779. Springer, Cham https://doi.org/10.1007/978-3-030-78221-4_1
- Bertola G (2021) BIM and rapid prototyping for architectural archive heritage. In: Ioannides M, Fink E, Cantoni L, Champion E (Hrsg) Progress in cultural heritage: documentation, preservation, and protection. EuroMed 2020. Lecture Notes in Computer Science, Bd. 12642. Springer, Cham https://doi.org/10.1007/978-3-030-73043-7_53
- Bischofberger WR, Pomberger G (1992) Prototyping-Oriented software development. Texts and Monographs in computer science. Springer, Berlin, Heidelberg <https://doi.org/10.1007/978-3-642-84760-8>
- Bjarnason E, Lang F, Mjöberg A (2023) An empirically based model of software prototyping: a mapping study and a multi-case study. *Empir Software Eng* 28:115. <https://doi.org/10.1007/s10664-023-10331-w>
- Carfagni M, Fiorineschi L, Furferi R, Governi L, Rotini F (2018) The role of additive technologies in the prototyping issues of design. *Rapid Prototyp J*. <https://doi.org/10.1108/rpj-02-2017-0021>
- Christoforakos L, Tretter S, Diefenbach S, Bibi SA, Fröhner M, Kohler K et al (2019) Potential and challenges of prototyping in product development and innovation. *i-com* 18(2019):179–187 (<https://pub.uni-bielefeld.de/record/2936802>)

- Croft BL, Clarkson JD, VonColln E (2019) User interface Prototyping Toolkit (UIPT). In: Marcus A, Wang W (Hrsg) Design, user experience, and usability. Design philosophy and theory. HCII 2019. Lecture Notes in Computer Science, Bd. 11583. Springer, Cham, S 195–207 https://doi.org/10.1007/978-3-030-23570-3_15
- Cruzes DS, Dybå T (2011) Recommended steps for thematic synthesis in software engineering. In: 2011 international symposium on empirical software engineering and measurement, S 275–284 <https://doi.org/10.1109/sem.2011.36>
- Da Silva Melo ÁH et al (2022) From requirements to prototyping: proposal and evaluation of an artifact to support interface design in the context of autism. In: Duffy VG, Gao Q, Zhou J, Antona M, Stephanidis C (Hrsg) HCI international 2022—late breaking papers: HCI for health, well-being, universal access and healthy aging. HCII 2022. Lecture Notes in Computer Science, Bd. 13521. Springer, Cham, S 307–321 https://doi.org/10.1007/978-3-031-17902-0_22
- Duc AN, Abrahamsson P (2016) Minimum viable product or multiple facet product? The role of MVP in software startups. In: Sharp H, Hall T (Hrsg) Agile Processes, in Software Engineering, and Extreme Programming. XP 2016. Lecture notes in business information processing, Bd. 251. Springer, Cham, S 118–130 https://doi.org/10.1007/978-3-319-33515-5_10
- El Khatib RE, Wrisley DJ, Elbassouni S, Jaber M, El Zini J (2019) Prototyping across the disciplines. Digit Stud 8(1):10. <https://doi.org/10.16995/dscn.282>
- Feldmann A, Teuteberg F (2019) From an idea to a prototype to a product ? Barriers after a co-creation event. ICIS 2019 Proceedings 8. https://aisel.aisnet.org/icis2019/future_of_work/future_work/8
- Fuhr L, Jurisch M, Böhm S, Braunmann T, Iglar B (2018) Pro2screen—high-fidelity-prototyping of mobile enterprise applications using process models. Int J Adv Intell Syst 11(1,2):127–133 (https://personales.upv.es/thinkmind/dl/journals/intsys/intsys_v11_n12_2018/intsys_v11_n12_2018_12.pdf)
- Gerasimov A, Michael J, Netz L, Rumpel B, Varga S (2020) Continuous transition from model-driven prototype to full-size real-world enterprise information systems. In: AMCIS 2020 Proceedings, Bd. 2 (https://aisel.aisnet.org/amcis2020/systems_analysis_design/systems_analysis_design/2)
- Gupta V, Bjarnason E, Gupta C (2022) Strategic prototyping technology adoption in startups: framework, challenges, and opportunities. IT Prof 24(3):88–95. <https://doi.org/10.1109/mitp.2022.3172876>
- Heil S, Siegert V, Gaedke M (2018) REWAMP: rapid web migration prototyping leveraging webassembly. In: Mikkonen T, Klamma R, Hernandez J (Hrsg) Web engineering. ICWE 2018. Lecture Notes in Computer Science, Bd. 10845. Springer, Cham, S 84–92 https://doi.org/10.1007/978-3-319-91662-0_6
- Hicking J (2020) Smart service Prototyping. In: Maleshkova M, Kühl N, Jussen P (Hrsg) Smart service management. Springer, Cham, S 63–74 https://doi.org/10.1007/978-3-030-58182-4_6
- Klippenstein A, Weskamp C, Laux F, Neuhaus F, Pfannenschmidt K, Bülling M et al (2023) A prototype to support business model innovation through crowdsourcing and artificial intelligence. Wirtschaftsinformatik 2023 proceedings 47. <https://aisel.aisnet.org/wi2023/47/>
- Kölbel T, Zekri A, Weinhardt C (2023) Developing blockchain-enabled marketplace interfaces: a design science research study. ICIS 2023 Proceedings. 1 (<https://aisel.aisnet.org/icis2023/blockchain/blockchain/1>)
- Lauff C, Kotys-Schwartz D, Rentschler ME (2018) What is a prototype? What are the roles of prototypes in companies?. ACME 2019. J Mech Des. <https://doi.org/10.1115/1.4039340>
- Levchenko A, Taratukhin V (2022) Challenges and prospects for cloud-based enterprise systems in tradition-focused cultures: a design thinking case study. In: Taratukhin V, Matveev M, Becker J, Kupriyanov Y (Hrsg) Information systems and design. ICID 2021. Communications in computer and information science, Bd. 1539. Springer, Cham, S 17–30 https://doi.org/10.1007/978-3-030-95494-9_2
- Leyh C, Wendt T (2018) Enterprise Systems als Basis der Unternehmens-Digitalisierung. HMD 55(1):9–24. <https://doi.org/10.1365/s40702-017-0389-z>
- Mayring P (2010) Qualitative Inhaltsanalyse. In: Mey G, Mruck K (Hrsg) Handbuch Qualitative Forschung in der Psychologie. VS, S 601–613 https://doi.org/10.1007/978-3-531-92052-8_42
- Pastoor S (2018) Phase 6: Prototyping – Ideen testen. In: Praxishandbuch Nachhaltige Produktentwicklung. Springer Gabler, Berlin, Heidelberg, S 189–206 https://doi.org/10.1007/978-3-662-57320-4_17
- Peffers K, Rothenberger M, Tuunanen T, Vaezi R (2012) Design science research evaluation. In: Peffers K, Rothenberger M, Kuechler B (Hrsg) Design science research in information systems. Advances in theory and practice DESRIST 2012. Lecture Notes in Computer Science, Bd. 7286. Springer, Berlin, Heidelberg, S 398–410 https://doi.org/10.1007/978-3-642-29863-9_29

- Plazas JE, Bimonte S, Schneider M, de Vaulx C, Corrales JC (2020) Self-service business intelligence over on-demand IoT data: a new design methodology based on rapid Prototyping. In: Darmont J, Novikov B, Wrembel R (Hrsg) *New Trends in Databases and Information Systems. ABDIS 2020. Communications in computer and information science*, Bd. 1259. Springer, Cham, S 84–93 https://doi.org/10.1007/978-3-030-54623-6_8
- Qiu S, Du L, Han T, Hu J (2018) Flavor explore: rapid prototyping and evaluation of user interfaces. In: Streitz N, Konomi S (Hrsg) *Distributed, ambient and pervasive interactions: understanding humans DAPI 2018. Lecture Notes in Computer Science*, Bd. 10921. Springer, Cham, S 114–123 https://doi.org/10.1007/978-3-319-91125-0_9
- Rupprecht D, Blum R, Bomsdorf B (2018) Software support for coherent prototyping of 3D gesture interactions. In: Bogdan C, Kuusinen K, Larusdottir M, Palanque P, Winckler M (Hrsg) *Human-centered software engineering. HCSE 2018. Lecture Notes in Computer Science*, Bd. 11262. Springer, Cham, S 207–218 https://doi.org/10.1007/978-3-030-05909-5_12
- Shakeel SI, Al Mamun M, Haolader M (2022) Instructional design with ADDIE and rapid prototyping for blended learning: validation and its acceptance in the context of TVET Bangladesh. *Educ Inf Technol* 28(6):7601–7630. <https://doi.org/10.1007/s10639-022-11471-0>
- Strangmeier R (2008) Was ist Wirtschaftsinformatik? – Zum Selbstverständnis einer jungen Wissenschaft, S 447–456 https://doi.org/10.1007/978-3-8349-9777-7_26 (Gabler eBooks)
- Vogel J, Schuir J, Koßmann C, Thomas O, Teuteberg F, Hamborg K (2021) Let's do design thinking virtually: design and evaluation of a virtual reality application. *ECIS 2021 Research Papers* (112). (https://aisel.aisnet.org/ecis2021_rp/112)
- Webster J, Watson RT (2002) Analyzing the past to prepare for the future: writing a literature review. *MISQ* 26(2) (<http://www.jstor.org/stable/4132319>)
- Wilde T, Hess T (2007) Forschungsmethoden der Wirtschaftsinformatik. *Wirtschaftsinformatik Angew Inform* 49(4):280–287. <https://doi.org/10.1007/s11576-007-0064-z>
- Wu CH, Chuang CN, Chang WY, Tsai WF (2018) Development of Big Data Multi-VM platform for rapid prototyping of distributed deep learning. In: Chin F, Chen C, Khan L, Lee K, Zhang LJ (Hrsg) *Big Data – BigData 2018. BIGDATA 2018. Lecture Notes in Computer Science*, Bd. 10968. Springer, Cham, S 182–193 https://doi.org/10.1007/978-3-319-94301-5_14
- Zhang YF, Sun L, Cao Q (2022) TLP-LDPC: Three-Level parallel FPGA architecture for fast prototyping of LDPC decoder using High-Level synthesis. *J Comput Sci Technol* 37(6):1290–1306. <https://doi.org/10.1007/s11390-022-1499-9>

Hinweis des Verlags Der Verlag bleibt in Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutsadressen neutral.