

Digitale Arbeitsumgebungen in der Produktentstehung

– Mit Action Design Research Web-Anwendungen zur produktiven Zusammenarbeit entwickeln.

Stephan Scheele, Daniel Mau, David Fouillois, Frank Mantwill

Um effektiv auf neue Marktgegebenheiten reagieren zu können, versuchen Industrieunternehmen ihre internen Geschäftsabläufe schlank und effizient zu halten. Dies stößt jedoch an Grenzen, wenn tradierte Unternehmens-IT im Zuge der digitalen Transformation mit neuen Geschäfts- und Bedienfähigkeiten ausgestattet werden soll. Die Integration von Systemen und Daten sowie die prozessuale Steuerung einer sich neu ordnenden IT-Landschaft verlangt nach neuen Konzepten, die die Besonderheit von kooperativen Wertschöpfungsprozessen berücksichtigen. Der Beitrag stellt die Anwendung der Action Design Research für die Konzeption, Entwicklung, Einführung und Auswertung einer IT-Applikation innerhalb eines Anwendungsfalls der Montageplanung der Automobilproduktion heraus. Dazu wird der Ansatz der Conversational Workflows, ein für die Zusammenarbeit in wissensintensiven, kooperativen Industrieprozessen entwickelter Applikationsaufbau, zur Lösung der Problemstellung herangezogen. Mit Hilfe eines auf den Anwendungsfall angepassten Ablaufs der Action Design Research werden in zwei partizipativen ADR-Zyklen sowohl eine MVP-Version als auch eine produktive Variante einer Web-Applikation für die Materialmodulbereitstellung entwickelt und im Praxissystem erlebbar gemacht. Im Ergebnis konnte sich das Konzept der Conversational Workflows als geeignet herausstellen und lässt zusätzlich Raum für kontinuierliche Weiterentwicklungen. Auf methodischer Seite hat sich gezeigt, dass eine Ergänzung der konzeptionellen ADR-Phasen um Design Thinking zu einer verbesserten Artefaktentwicklung beitragen kann.

Keywords: Enterprise Apps, Action Design Research, Collaboration, Design Thinking, Conversational Workflow Management

Einleitung und Hintergrund

Die informationstechnologische Repräsentation von Geschäftsprozessen mit Hilfe von Workflows ist seit ca. 25 Jahren Stand der Technik und ein wichtiger Baustein in der

Entwicklung und Planung von Enterprise IT-Systemen. Die Abbildung wertschöpfender Anteile der Arbeit durch miteinander verbundene Geschäftsprozesse kann als ein zentrales Element der betrieblichen Leistungserstellung gesehen werden (Scheele & Mantwill: 2019a). Dies spielt insbesondere in der Prozesslandschaft der industriellen Produktentstehung eine Rolle, da vor allem dort im Rahmen des Produktentstehungsprozesses wertschöpfende Arbeit entlang der Prozesskette geleistet wird. Da diese Arbeit aufgrund von Spezialisierung und Umfang nicht von einzelnen Mitarbeitenden isoliert verrichtet werden kann, spricht man von Zusammenarbeit oder auch Kollaboration. Kollaboration beschreibt die Arbeit von mindestens zwei Individuen an gemeinsamen Materialien, die auf ein gemeinsames Gruppenziel ausgerichtet ist. Zur Erreichung des Gruppenziels sind Kommunikation, Koordination und Kooperation notwendig (Leimeister: 2014).

Eine wesentliche Ausprägung der Kollaboration ist die gemeinsame Arbeit am jeweiligen Produkt oder dessen Teilelementen. Diese Arbeit ist Teil mitanderer verketteter Geschäftsprozesse, wie bspw. Übergabe von Arbeitsständen, Freigabeprozessen, Lieferanteneinbindungen, Datenmeldungen oder Systemübertragungen. Die einzelnen Elemente der Arbeit, die Aufgaben, werden grundsätzlich dem Bereich der Wissensarbeit zugeordnet, bei dem die Mitarbeitenden in ihrem jeweiligen Aufgabengebiet ihre Arbeitsleistung unter Verwendung von Fachwissen, Erfahrungswissen oder kognitiven und problemlösenden Verarbeitungsprozessen einbringen. Dem gegenüber steht Routinearbeit, bei der lediglich einfache, wenig komplexe Tätigkeiten verrichtet werden. (Scheele & Mantwill: 2019b)

Aus dem Blickwinkel einer softwaretechnischen Unterstützung dieser Arbeitsprozesse in der Produktentstehung haben sich entlang der Prozesskette und je nach Phase der Produktreife unterschiedliche IT-Systeme etabliert. Diese IT-Systeme lassen sich in zwei übergeordnete Kategorien einteilen, zum einen die *Autorensysteme*, zum anderen die *Verwaltungssysteme*. Autorensysteme erzeugen Daten in Form von Geschäftsobjekten, zu denen bspw. CAD-Bauteile, Simulationsergebnisse oder digitale Fabrikmodelle gehören können. Verwaltungssysteme hingegen benutzen, steuern und orchestrieren diese Geschäftsobjekte, indem sie entweder logische Sammler (Projekte) oder Ablage-logiken (Produktstrukturen, Stücklisten etc.) aufbauen. Die Geschäftsobjekte werden in ihrem originären Datenformat oder in diversen Austauschformaten eingelesen und können im Zugriff der jeweiligen Verantwortlichen in ihrer Prozessreife verwaltet, mit Metadaten annotiert oder modifiziert werden. Prominentes Beispiel dieses Zusammenspiels ist die Verwendung von CAD-Systemen (Autorensystem) und Produktdatenmanagement-Systemen (PDM). (Gerhard: 2016)

Wird nun eine Zusammenarbeit zwischen einzelnen Beteiligten in Geschäftsprozessen angestrebt, wie bspw. bei der angesprochenen Bauteilfreigabe inklusive der Dokumentation in Produktstruktur und Stückliste, setzen die etablierten Systeme auf systeminterne Workflow-Technologien. Das jeweilige Workflow-Objekt lässt sich somit im definierten Arbeitsablauf den beteiligten Rollen ausspielen. Dies stößt jedoch an Grenzen, sobald system- oder gar domänenübergreifende Prozesse bedient werden müssen (Scheele & Mantwill: 2019b). Interne Workflows können das System als geschlossenes „Silo“ kaum verlassen oder es lassen sich für den Prozess wichtige Geschäftsobjekte nicht zwischen Systemen effizient austauschen. Dieser Aspekt ist jedoch häufig notwendig. Neuartige Workflow- und Work-Management-Lösungen, wie sie bspw. im privaten oder semi-privaten Umfeld eingesetzt werden, überzeugen zwar mit sehr guter Gebrauchstauglichkeit und einem guten Nutzererlebnis, lassen sich jedoch schwerlich in die Enterprise IT einbetten (Schnittstellen im Sinne des Zugriffs auf originale Daten tradierter Verwaltungssysteme) und scheitern oftmals an Vorgaben zum Datenschutz und Datenspeicherort. Im Ergebnis zeigt sich eine Eigenverwaltung der Prozessdaten mittels Standard-Bürosoftware auf Flughöhe fachbereichsbezogener Selbstoptimierung. Oftmals wird dies ausgeführt als sog. „Schatten-IT“ (Urbach, Ahlemann: 2016).

Gleichzeitig zeigen jüngste Entwicklung auf dem Gebiet der Enterprise IT, dass Industrieunternehmen im Rahmen ihrer nach innen gerichteten Digitalisierungsbemühungen vermehrt auf den internen Einsatz von Cloud-Software und Integrationsplattformen großer Infrastrukturanbieter setzen, wie bspw. Amazon Web Services oder Microsoft Azure (sog. „Hyperscaler“). Somit entsteht eine Stoßrichtung, die vermuten lässt, dass die datenhaltenden Systeme sich sukzessive öffnen oder durch die Anbindung an Integrationsplattformen geöffnet werden. Gepaart mit der durch privaten Konsum befeuerten Erwartungshaltung der Nutzer an höhere Gebrauchstauglichkeit betrieblicher Software zeigen sich Entwicklungen, die den Einsatz moderner Anwendungssoftware mit entsprechenden Datenversorgungstechnologien, wie Event Streaming, semantischer Vernetzung oder Schnittstellen-Ökosystemen beschleunigen.

Im vorliegenden Beitrag werden am Beispiel eines Anwendungsfalls aus der Automobilindustrie die Ergebnisse einer Studie zur Entwicklung und Implementierung einer Enterprise Web App im industriellen Umfeld der Produktentstehung präsentiert. Bei dem in der Studie betrachteten Partnerunternehmen handelt es sich um den deutschen Automobilhersteller OEM. OEM ist Teil des Mehrmarkenkonzerns Auto AG. Die Auswahl der Automobilindustrie als Forschungsobjekt begründet sich zum einen in der Tatsache, dass (1) die Automobilindustrie in der Komplexität und Frequenz der

Produktentstehung in einer Spitzenposition zu finden ist, zum anderen (2) die gewünschten positiven Effekte erfolgreicher Digitalisierungsbemühungen trotz hoher Investitionen in den Ausbau und die Weiterentwicklung der IT-Systeme auszubleiben scheinen und letztlich (3) die angesprochenen Cloud-Plattformen zunehmend aufgebaut werden, die eine moderne und effektive Softwareentwicklung zur Prozessunterstützung erst ermöglichen. Konkret handelt es sich bei der Studie um die Entwicklung einer Funktionalität zur Bestellung und Lebenszyklusverwaltung von Materialmodulen für die Materialanstellung der Fahrzeugmontage im Rahmen der Produktionsplanung.

Als Grundgerüst der Prozess- und Softwareentwicklung wird das Konzept der Conversational Workflows herangezogen, ein vom beheimateten Institut der Autoren entwickelter Ansatz zur Abbildung und Steuerung kooperativer Arbeitsprozesse in komplexen industriellen Zusammenhängen. Als Forschungsmethodik wird die Action Design Research (ADR) nach Sein et al. (2011) verwendet, einer Kombination der Design Science Research (DSR) mit der sozialwissenschaftlichen Aktionsforschung. ADR scheint geeignet zu sein, innovative IT-Artefakte partizipativ zwischen Wissenschaft und Praxis zu gestalten sowie in organisatorischen (i.d.R. unternehmerischen) Kontexten zu verproben. Im Rahmen der Studie wurden folgenden zwei Forschungsfragen verfolgt:

- (FF1) Lassen sich die Design Prinzipien des Ansatzes der Conversational Workflows auf weitere Anwendungsfälle der Produktentstehung übertragen und lässt sich das darin gebundene Designwissen somit verallgemeinern?
- (FF2) Wie lassen sich mit Hilfe von Design Thinking die präskriptiven Phasen der ADR methodisch stärken und somit konkretisieren?

Der erste Teil des Beitrags stellt die Methode Action Design Research vor (Kap. 2) und präsentiert die konkrete Ausgestaltung im Rahmen eines Praxisprojektes (Kap. 3). Der zweite Teil liefert die inhaltliche Ebene, in dem aufbauend auf dem Konzept der Conversational Workflows die Ergebnisse des ADR-Projektes dargelegt werden (Kap. 4). Zusätzlich wird eine Erweiterung der Methode ADR um Elemente des Design Thinkings vorgeschlagen (Kap.5). Zum Schluss findet eine kritische Würdigung sowohl des entstandenen IT-Artefaktes für die Praxisvertreter und Endnutzer als auch die Qualifizierung der Conversational Workflows für diesen Anwendungsfall statt.

Action Design Research

Entstehung, Phasen und Prinzipien

Im Fachgebiet *Information Systems*, dem amerikanischen Pendant der deutschsprachigen Wirtschaftsinformatik, wird seit mehreren Jahren ein Richtungsstreit zwischen Anhängern der gestaltungsorientierten Forschung und jenen der historisch dominierenden, verhaltensorientierten Ausrichtung geführt. Den Anhängern der gestaltungsorientierten *Design Science Research* (vgl. Designwissenschaft; künstlich, von Menschen gemachte, im Gegensatz zu Natur- und Sozialwissenschaften (Simon: 1995)) wird eine fehlende Rigorosität und Abgrenzbarkeit zu kommerzieller Beratung und Softwareentwicklung vorgehalten (Hevner et al.: 2004). Dem gegenüber stehen wegweisenden Beiträge (March & Smith: 1995; Hevner: 2007; Hevner & Chatterjee: 2010; Österle et al.: 2010; Gregor & Hevner: 2013; Vaishnavi & Kuechler: 2019), die eine höhere Praxisrelevanz und Gestaltungsorientierung der Fachrichtung fordern. Mittlerweile gilt die DSR als beliebtes und anerkanntes Rahmenwerk für anwendungsnahe, praxisrelevante Forschung rund um die Konstruktion von betrieblichen Informationssystemen und informationstechnologischen Artefakten, wie sie im Zuge der allgemeinen Digitalisierung häufig Gegenstand von Forschungsvorhaben sind (Robra-Bissantz & Strahringer: 2020).

Obwohl die DRS auf die Konzeption, Entwicklung und Implementierung von innovativen IT-Artefakten zur Lösung von realen, tatsächlichen und verallgemeinerbaren Problemen abzielt („[...] *real world problems* [...] (Hevner et al.: 2004)), wird kritisiert, dass sie nicht zwangsläufig einen definierten (organisatorischen) Kontext berücksichtigt (Redlich et al.: 2020). Ebenfalls wird kritisiert, dass die für die DSR gängigen Ablaufschemata, wie bspw. die Design Science Research Methodology nach Peffers (2008), einem zu starren Stage-Gate-Prozess folgen und Praxisinput und gemeinsames Erarbeiten der Lösung zusammen mit Praxisvertretern zu wenig unterstützen. Zwar wird dies durch die Einbeziehung des Praxissystems in der DSR bereits als wesentlich hervorgehoben (Hevner & Chatterjee: 2010), jedoch bleiben Praxisvertreter neben der Institution als solche in der Rolle der Forschungsobjekte. An dieser Stelle setzt die Action Design Research an. Sie kombiniert die DSR mit der Aktionsforschung, einer Methode, bei der Forscher und Praktiker (Stakeholder des Projektes) gemeinsam an der Entwicklung des Artefakts arbeiten (Sein et al.: 2011; Redlich et al.: 2020). Auf diese Weise entsteht das sog. Ensemble-Artefakt, eine IT-basierte Entwicklung in einem sozial-organisatorischen Kontext, was sowohl die Gegebenheiten des jeweiligen Kontexts berücksichtigt als auch durch die partizipative Entwicklung an Reife gewinnt.

Ablauf der Action Design Research

Sein et al. (2011) definieren in vier übergeordneten Phasen eine Ablauflogik der ADR (siehe Abbildung 1). Die erste Phase (*Problem Formulation*) dient der Identifizierung und Diagnostik von realen Problemen „im freien Feld“ und bietet Forschenden die Gelegenheit, in den Forschungsprozess einzusteigen (*Knowledge Creation Opportunities*).

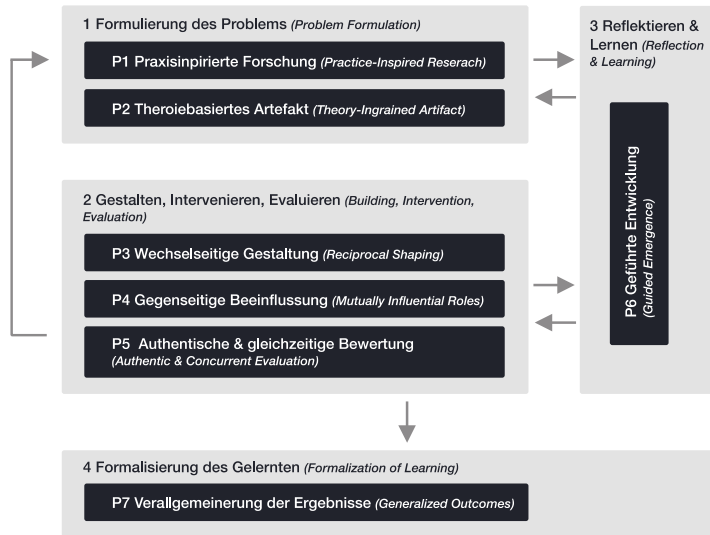


Abbildung 1: Phasen und Prinzipien der Action Design Research nach Sein et al. (2011), übersetzt

Die zweite Phase der ADR stellt das eigentliche Kernelement der Methode dar (siehe Abb. 2). Der sog. BIE-Zyklus (*Building, Intervention, Evaluation*) beinhaltet die Prinzipien der gemeinsamen, wechselseitigen Gestaltung, gegenseitigen Beeinflussung und der gleichzeitigen, authentischen Bewertung (Sein et al.: 2011).

Reflektion und Lernen sind die Überschriften der dritten ADR-Phase. Das Prinzip der *guided emergence* dient in Parallelisierung der ersten beiden Phasen zur Detaillierung des Problem- und Lösungsraums. Die vierte und letzte Phase beschäftigt sich mit der Formalisierung des Gelernten. Zentraler Aspekt dabei ist es, die situativen Erkenntnisse insoweit in Lösungskonzepte zu überführen, sodass diese für ähnliche Probleme als mögliche Lösung dienen können („[...] *the situated learning from an ADR project should be further developed into general solution concepts for a class of field problems.*“ (Sein et al.: 2011)).

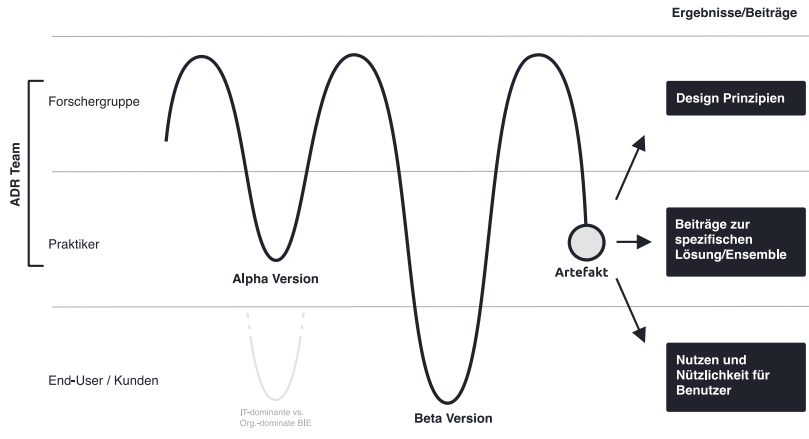


Abbildung 2: Zyklen der iterativen BIE-Phase nach Sein et al. (2011), übersetzt, angepasst

Vorgehensmodell

In Anlehnung an die Phasen von Sein et al. (2011) wurde das Projekt in zwei Zyklen unterteilt, Alpha und Beta. Die Detaillierung in Abb. 3 folgt einem von den Autoren entwickelten Ablaufmodell, was bereits bei der ursprünglichen Entwicklung der Conversational Workflows angewendet worden ist und auf einem Vorschlag von Schacht, Morana & Mädche (2015) beruht. Die Reflektion des Alpha-Zyklus bildet gleichzeitig die Grundlage für den folgenden Beta-Zyklus, sodass erzielte Erkenntnisse und Lerneffekte in die neue Iteration übernommen werden können. Personell wurde ein ADR-Team aufgestellt, was aus der Forschergruppe und Vertretern von OEM bestand.

Bei den Praktikern handelte es sich um Vertreter der Prozess- und Methodenentwicklung der Produktionsplanung von OEM von zwei Werksstandorten sowie eines Vertreters der IT-Abteilung. Die Endkunden der Software sind die operativ tätigen Produktionsplaner und Materialmodulhersteller.

Gestartet wurde das Vorhaben mit der Problembeschreibung und Bedarfsformulierung des Unternehmens OEM (*practice inspired research*). Daraufhin hat das Forscherteam eine erste empirische Studie (Status Quo Analyse) im Unternehmensumfeld durchgeführt. Maßgebliche Analyseformen waren qualitative Interviews/Experteninterviews mit Beteiligten und Betroffenen, hospitierende Beobachtungen und die Recherche von unternehmensinterner Dokumentation. Nach erfolgter Datenanalyse und

einer Prüfung der identifizierten Probleme, Bedarfe und Lösungsfelder gegen bekannte Theorien mit entsprechendem deskriptivem und präskriptivem Wissen (*theory ingrained artifact*) wurde zusammen mit den Praktikern eine erste Version der Lösung als *Minimal Viable Product* (MVP) konzipiert, entwickelt (*reciprocal shaping*) und mit den Anforderungen und Erwartungen der Endkunden abgeglichen. Nach erfolgter Evaluation der ersten Version inklusive Erkenntnisableitung innerhalb des ADR-Teams und erneuter Analyse folgte der Beta-Zyklus in analoger Abfolge (Artefakt 2).

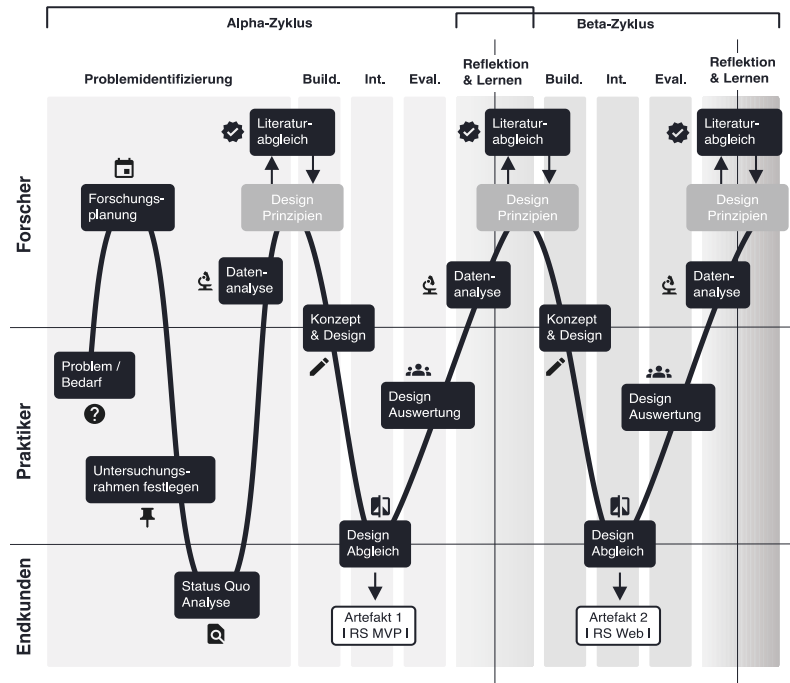


Abbildung 3: Forschungspfad in Anlehnung an die ADR-Phasen
 Horizontale Achse: Zeit (nicht proportional). Vertikale Achse: Forscher arbeiten exklusiv,
 Forscher arbeiten mit Praktikern, Forscher arbeiten mit Endkunden.

Die Aufgabe der Forscher lag neben dem Prozess *Building, Intervention, Evaluation* darin, einen immer wiederkehrenden Abgleich des Gelernten mit der bekannten Wissensbasis durchzusetzen. Somit entstehen die von Sein et al. (2011) vorgetragenen Ergebnistypen auf Basis der unterschiedlichen Sichten auf das Ensemble-Artefakt: Aus Sicht der Endnutzer die jeweiligen Artefakte als konkrete und nützliche IT-Lösungen, für die Praktiker die durch das Forschungsprojekt unterstützte und begleitete Artefakt-

Entwicklung mit der Annahme eines zusätzlichen Qualitätsgewinns und letztlich für die Forscher die Rückführung von deskriptivem und präskriptivem Wissen in Form der Designprinzipien.

Materialmodulverwaltung in der Produktionsplanung

Conversational Workflow Management

Das Prinzip der Conversational Workflows fußt auf der Vision einer intelligenten IT-Unterstützung für Wissensarbeiter in komplexen, kooperativen Arbeitsabläufen. In Anlehnung an eine einfache und intuitive Kollaborationsform durch Nachrichten in Messenger-Software wird diese Idee um eine Prozesskomponente erweitert: Arbeitsabläufe mit mehreren Beteiligten werden als interaktive Workflows modelliert, bei denen jeder seine individuelle Aufgabe als dezidierte Aktion ausgespielt bekommt. Innerhalb dieser Aktion befinden sich alle für die Aufgabe notwendigen Informationen zu Geschäftsobjekten, die ggf. aus unterschiedlichen Quellsystemen stammen können. Je erfolgreicher eine Aktion ist, desto mehr Informationen werden in der Historie des Workflows als solches erhalten. Aktionen können beliebig komplexe Inhalte darstellen, die in ihrer Spezifität den Charakter von „Micro-Apps“ besitzen. Im Gegensatz zur konventionellen Enterprise App-Entwicklung bleibt eine solche Micro-App nicht fachlich isoliert, sondern versucht durch den Prozesskontext eine höhere Transparenz und eine bessere Prozessführung zu bieten. Workflows lassen sich je nach Bedarf organisieren, quervernetzen und in Teilschritten automatisieren, um komplexe Zusammenarbeit in der Produktentstehung einfach und effizient abbilden zu können.

ADR Zyklus 1 – Entwicklung, Intervention und Evaluation

Die Materialmodulverwaltung ist ein wichtiger Teilbereich der Montageplanung beim Automobilhersteller OEM. Bereits Jahre vor Anlauf einer Fahrzeugproduktion werden Montagestraßen geplant und verschiedene Planvarianten hinsichtlich Kennzahlen wie Durchsatz, Zeit, Taktung und Kosten vergleichbar gestellt. Als Materialmodule gelten unterstützende Betriebsmittel wie Vor- und Rücklaufregale, Sequenzwagen, Tische, Ablagen etc. Die Module werden mehrheitlich in Eigenfertigung mittels Rohrsteckverfahren gebaut und ggf. mit Ausstattungen wie *Pick-by-Light*-Technik aufgerüstet. Der Prozess für die Bestellung und Verwaltung solcher Module wurde in der Vergangenheit im Wesentlichen über Mail, Excel-Listen und Outlook-Aufgaben gesteuert und führte regelmäßig zu Doppelbeauftragungen, fehlender Kontrolle und geringer Transparenz. Hinzu kam, dass einige der Materialmodulinformationen bereits in anderen IT-Systemen

gepflegt wurden, dazugehörige Fahrzeugprojekte und Planvarianten in weiteren Systemen, die ebenfalls bedient werden mussten. Zusammenfassend gab es aus Sicht der Praxis drei Motivationen, den Prozess der Materialmodulverwaltung zu renovieren:

- Integrierte Regalbaudatenbank: Keine bestehende Möglichkeit, Materialmodule und Ressourcen zu definieren, zu verfolgen und in den Kontext der Planungsaktivitäten zu setzen.
- Transparenz und Statuskontrolle: Verantwortliche für die Materialanstellung benötigen Überblick und Kostenkontrolle.
- Datenseitige Eindeutigkeit: Wiederverwendung bereits vorhandener Geschäftsobjekte, *Single Point of Truth* in der Nutzung von systemübergreifenden Daten.

Prozessseitig ist eine Zusammenarbeit zwischen verschiedenen Rollen notwendig, die jeweils unterschiedliche Aufgaben haben. Der sog. Takter meldet den Bedarf an Modulen. Der Teilprojektleiter bestätigt den Bedarf und erteilt Bauaufträge. Der Fachprojektleiter ist für die Budgetverteilung, Kostenüberwachung und Flächenplanung verantwortlich. Die Werkstätten bewerten Kosten, fertigen das Modul, verantworten die Logistik und die Reparatur. Der Betreiber bestätigt das Modullayout, kontrolliert den Einsatz und verantwortet die turnusmäßige Sicherheitsprüfung.

Im Rahmen der Status Quo Analyse und der Datenanalyse wurden entsprechende Inhalte aufgenommen und für die anstehende Entwicklungsphase mit den Design Prinzipen der Conversational Workflows und weiteren Literaturquellen aus dem Bereich Softwareentwicklung und Interface Design abgeglichen. Auf Basis dessen wurde ein Vorschlag in Form einer Web-Applikation mit dem Namen *Rack Shop* unterbreitet, die den Ansatz der Conversational Workflows implementiert. Alle beteiligten Rollen werden in einem interaktiven *Feed* zusammengeführt und erhalten rollenbasiert ihre jeweilige Aktion ausgespielt. Jeder Workflow entspricht einem Bestellvorgang eines Materialmoduls. Neben den Workflows werden Geschäftsobjekte (Module, Bauaufträge, Kostenvoranschläge etc.) gelistet und deren Informationen zusätzlich dargestellt.

Auf technischer Seite wurde die Anwendung unter Verwendung fortschrittlicher Open Source Web Technologien (*React.js*, *Node.js*, REST, *Material*) entwickelt und mittels der Web-Emulation *Electron* als installierbare Desktopanwendung ausgeliefert. *Electron* basiert auf der *Chromium*-Engine und paketierte Web Code. Der Vorteil dieses Ansatzes ist, dass nahezu gleicher Programmiercode zur Web-Implementierung verwendet werden kann und sich die Anwendung unkompliziert auf den Rechnern der jeweiligen Testanwender ausführen lässt. Die Datenhaltung findet auf einem Netzlaufwerk in gemeinsamen Zugriff statt.

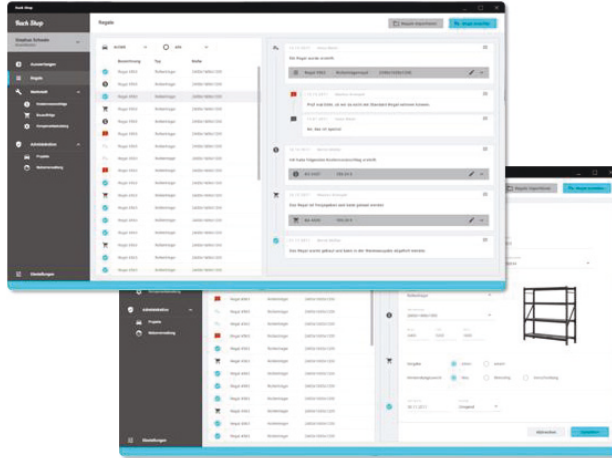


Abbildung 4: Minimal Viable Product der Anwendung Rack Shop

In der Evaluation hat sich herausgestellt, dass bereits in der ausgewiesenen Testphase Teile eines Fahrzeugprojektes über die Anwendung abgebildet worden sind, was einer Verifizierung des Artefaktes durch dessen Anwendung nahekommt. Auch wurde das Artefakt während des kompletten Designprozesses von den Praktikern und Endnutzern kontinuierlich evaluiert. Die Evaluation stützte sich auf Feldbeobachtungen, informelle Interviews, Nutzungsstatistiken und die aktive Beteiligung im Scrum-Prozess. Im Ergebnis war die Rückmeldung der Endnutzer aller Rollen positiv, sodass umfangreiche fachliche Erweiterungswünsche aufgekommen sind, was in der Gruppe der Forscher zu einer Erweiterung der Design Prinzipien und der Vorbereitung des zweiten Zyklus führte.

ADR Zyklus 2 – Entwicklung, Intervention und Evaluation

Die fachliche Weiterentwicklung des *Rack Shops* bezog sich insbesondere auf eine Hinzunahme weiterer Workflows und Materialmodule. Wo in der MVP-Version lediglich Module des Typs Regal im Workflow der Bestellung abgebildet wurden, sollte nun der komplette Lebenszyklus betrachtet werden. Dies inkludiert Workflows wie umbauen, umziehen, einlagern, recyceln, prüfen, reparieren und Defekte melden, jeweils mit ähnlicher, rollenbasierter Steuerung. Auch wurde ein umfassendes Dashboard- und Berichtswesen und eine Budget- und Vorschauplanung auf Taktebene entwickelt.

Technisch konnte die Applikation in eine Enterprise Cloud-Umgebung nach dem Prinzip *Platform-as-a-Service* überführt werden und ist dort unter eigener API und für autorisierte Benutzer ortonabhängig verfügbar. Zusätzlich wurde eine Anpassung im Sinne des *Responsive Designs* vorgenommen, was eine Ausführung auf verschiedenen Endgeräten ermöglicht. Dies ist bspw. bei Personal im Feld wichtig, da diese Mitarbeiter oft mit Tablets oder Smartphones ausgestattet sind.

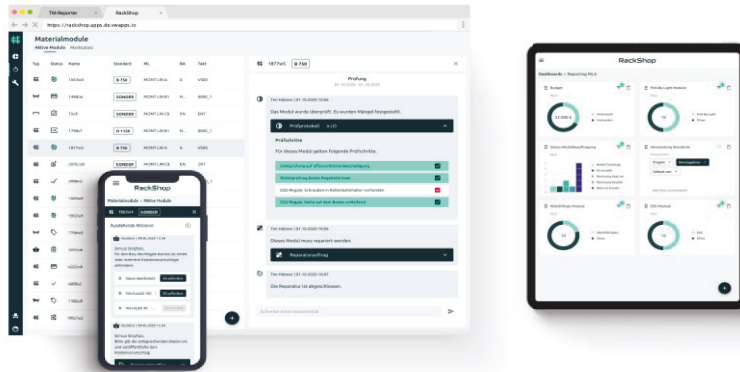


Abbildung 5: Web-Applikation Rack Shop, Multi Device

Auch hier haben die Phasen der Intervention und Evaluierung zu einer Reihe an fachlichen Weiterentwicklungsideen geführt, die sowohl zur Anpassung der Design Prinzipien als auch für zukünftige ADR-Zyklen aufgenommen worden sind. Die Evaluations-techniken des ersten ADR-Zyklus wurden größtenteils beibehalten, wenn gleich der Kreis der Stakeholder größer war. Bspw. wurde das Artefakt in mehreren Managementkreisen vorgestellt; auch wurde der Blickwinkel der Beurteilung erweitert, in dem IT-architektonische und fachliche Entscheiderrunden mit eingebunden wurden, wie bspw. Werkstätten und operative Montageplanung. Zusammenfassend wurden die ADR-Prinzipien *wechselseitige Gestaltung*, *gegenseitige Beeinflussung* und *authentische & gleichzeitige Bewertung* verfolgt.

Optimierung der ADR-Phasen durch Design Thinking

Die Ausgestaltung der ADR bietet Freiraum zur Integration verschiedener Frameworks und Einzelmethoden, da die Autoren um Sein et al. (2011) recht vage bleiben, *wie genau* die einzelnen Phasen konkretisiert werden können. Keijzer-Broers & de Reuver (2016)

argumentieren, dass aufgrund begrenzter Ressourcen ADR-Forscher effiziente Methoden einsetzen sollten, um ihre Designstrategie zu steuern, ohne dabei die beabsichtigten Forschungsziele aus den Augen zu verlieren. Um kontinuierliche Anpassungen der Problemstellung und damit einer durchgehenden Anforderungsänderung während des Entstehungsprozesses Rechnung zu tragen und schnelle Iterationen zu ermöglichen, erscheinen in der Praxis vor allem agile Arbeitsweisen, die auf Flexibilität, Anpassungsfähigkeit und Produktivität setzen, vielversprechend (Keijzer-Broers & de Reuver: 2016).

Während in der einschlägigen Literatur agile Methoden zu weiten Teilen synonymisch zum Scrum-Begriff verwendet werden, mangelt es diesem Ansatz an einer ausreichenden Betrachtung der konzeptionellen Phase (Dybå & Dingsøy: 2008). Zur Überwindung dieser Limitation gibt es Bemühungen weitere Methoden, insbesondere Design Thinking (DT), in die Software-Entwicklung zu integrieren und die interdisziplinäre Zusammenarbeit zu fördern (Hildenbrand & Meyer: 2012; Lindberg, Meinel, & Wagner: 2011). DT wird hier üblicherweise als Mittel der Anforderungsanalyse und -erhebung vor der eigentlichen agilen Entwicklung eingesetzt (Dobrigkeit & de Paula: 2017).

In der Literatur wird DT ursprünglich als human-zentrierter Innovationsansatz beschrieben, dessen Kern ein systematisches und kollaboratives Vorgehen zur Identifikation und kreativen Lösung von Problemen ist (Brown; 2008; Grots & Pratschke: 2009; Luchs, Swan, & Griffin: 2016). Grundlegend ist die Idee, iterativ eine Lösung in engem Austausch mit Interessengruppen und Zielanwendern zu entwickeln, um Wünschbarkeit, Durchführbarkeit und Machbarkeit der endgültigen Lösung sicherzustellen. Die Methodik bietet einen Prozessrahmen, der eine kontinuierliche Kommunikation zwischen den Entwicklern, Stakeholdern und Zielanwendern ermöglicht. (Vetterli et al.: 2013) Das zugrundeliegende Prozessmodell wird in der Literatur je nach Herkunft in unterschiedliche Phasen gegliedert. Nachfolgend soll das in der Wissenschaft verbreitete fünfphasige Modell nach Ideo (*Emphasize, Define, Ideate, Prototype, Test*) verwendet werden. Bezogen auf den konzeptionellen Teil der Aufgabenstellung sollen nachfolgend Ansatzpunkte zur Integration von DT in die Phasen des ADR vorgeschlagen werden. Zunächst zeigt sich durch die auf abstrakter Ebene übereinstimmend formulierte Triangulation von Wissenschaft (Entwicklung), Industrie (Stakeholder) und Endkunden (Zielanwender) eine ganzheitlich mögliche Integration von DT ins ADR. Unter Zugrundelegung von Mikro- und Makroprozessen (Brenner & Uebernickel: 2016) eines DT Projekts und der zweidimensionalen Trennung von divergenten und konvergenten Denkphasen entsteht außerdem die Möglichkeit der kontinuierlichen Ergänzung der einzelnen ADR Phasen um DT Phasen. Unter Voraussetzung eines gemeinsam gelebten

Mindsets (Redlich & Rehtien & Schaub: 2019) entsteht somit ein Ansatz zur ganzheitlichen prozessualen und mentalen Ausgestaltung.

In der Phase der Problemformulierung ermöglicht dieser Ansatz durch eine holistische und interdisziplinäre Betrachtung des Problemraums (*Emphathize*) mit darauf aufbauender Erkenntnisableitung (*Define*) methodische Unterstützungspotenziale. Die iterative Identifizierung der Forschungslücke und Entwicklung der Forschungsfragen wird damit auch der Forderung nach einer *Practice-Inspired Research* und einem *Theory-Ingrained Artifact* gerecht. Die anschließenden BIE-Phasen befördert der humanzentrierte DT Ansatz, durch die kontinuierliche Integration der Stakeholder und Positionierung der Kundenbedürfnisse, in den Mittelpunkt der Lösungsentwicklung (*Mutual Influential Roles*). Die Prozessphasen *Ideate*, *Prototype* und *Testing* gewährleisten eine methodische Unterstützung zur sukzessiven Entwicklung (*Reciprocal Shaping*) des Ensemble-Artefakts. Die Anzahl der Iterationen hängt direkt mit der Konkretisierung des Artefakts zusammen und ist je nach Projekt unterschiedlich (*Authentic & Concurrent Evaluation*) (Becker et al.: 2020). Abbildung 6 zeigt die mit zunehmender Projektdauer fortschreitende Konkretisierung des Ensemble-Artefakts. Die Integration der Makro- und Mikroprozess-Ebene ermöglicht auch ein iteratives Zurückspringen bis in die Problemformulierung bei grundlegender Anforderungsänderung.

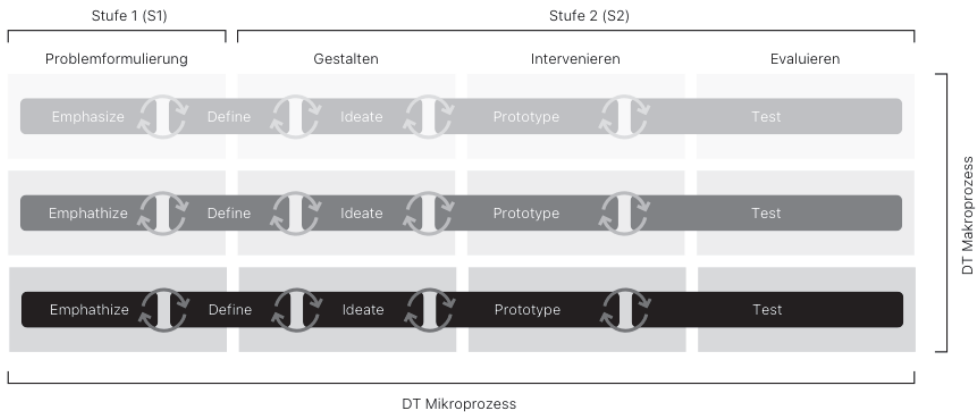


Abbildung 6: Integration von Design Thinking in die Action Design Research

Fazit und Ausblick

In Summe liefert der Beitrag unterschiedliche Ergebnisse. Zum einen konnte ausgehend von einer konkreten fachlichen Problemstellung aus der Produktionsplanung

des Unternehmens OEMs eine für die Anwender nützliche Applikation in Form eines IT-Artefaktes bereitgestellt werden, was einer relevanten Lösung nach den Prinzipien der gestaltungsorientierten Wirtschaftsinformatik entspricht. Auf methodischer Ebene hat sich die Anwendung der Action Design Research als Rahmenwerk für praxisnahe Forschungsvorhaben als passfähig und geeignet darstellt. Auf konzeptioneller Ebene konnte das Konzept der Conversational Workflows in der konkreten Anwendung zum Einsatz gebracht und in seinen Design Prinzipien weiterentwickelt werden, was einer Erweiterung der Wissensbasis für Forschende und einer Generalisierung der Ergebnisse entspricht.

Zur weiteren methodischen Ausgestaltung der ADR wurde vorgeschlagen, das Design Thinking als human-zentriertes mentales und prozessuales Framework zu verwenden. Dabei ist besonders die Eignung für den konzeptionellen Anteil des Ensemble-Artefaktes hervorzuheben. Die prozessbedingt fest verankerten Iterationen unter Einbezug aller Stakeholder fördern die Entwicklung von Antworten auf komplexe Fragestellungen. Hier zeigt sich das synergetische Potenzial. Aufgrund der Novität des gesamten Forschungsbereichs ist die methodische Ausgestaltung einzelner Teilbereiche des ADRs durch Design Thinking in der Literatur bisher wenig diskutiert. Offen bleiben daher konkretere Anforderungen an die Rahmenbedingungen und die methodische Umsetzung auf operativer Ebene. Hier bedarf es weiterer Forschungsarbeit.

Ebenfalls bedarf es weiterer Forschungsarbeit in der Begleitung und Verbesserung des Ansatzes der Conversational Workflows. Durch die Neuartigkeit sind längerfristige Einsatzbeobachten noch nicht möglich, sodass sich die These der verbesserten Zusammenarbeit durch Conversational Workflows in unterschiedlichen Anwendungskontext weiter beweisen muss. Zusätzlich gilt es, das Konzept robuster, generischer und schneller anpassbar für beliebige Szenarien zu machen.

Literaturverzeichnis

Becker, Felix et al. 2020. Open KMU: Mit Action Design Research und Design Thinking gemeinsam innovieren. *HMD Praxis der Wirtschaftsinformatik* 57, 2, 274–284.

Brenner, Walter & Uebernickel, Falk 2016. *Design Thinking for Innovation: Research and Practice*. Springer Heidelberg.

Brown, Tim 2008. *Design Thinking*. *Harvard Business Review*, 86, 6, 85–92.

Dobrigkeit, Franziska & de Paula, Danielly 2017. The best of three worlds - The creation of innodev a software development approach that integrates design thinking, scrum and lean startup. *Proceedings of the International Conference on Engineering Design, ICED 8 (DS87-8)*, 319–328.

- Dybå, Tore & Dingsøy Torgeir 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50, 10, 833–859.
- Gerhard, Detlef 2016. Daten- und Informationsmanagement PDM/PLM. In U. Lindemann, hg. *Handbuch Produktentwicklung*. München: Hanser, 215–245.
- Gregor, Shirley & Hevner, Alan R. 2013. Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly* 37, 2, 337–355.
- Grots, Alexander & Pratschke, Margarete 2009. Design Thinking – Kreativität als Methode. *Marketing Review* St. Gallen, 26, 2, 18-23.
- Hevner, Alan R. 2007. A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems* 19, 2
- Hevner, Alan R. et al. 2004. Design Science in Information Systems Research. *MIS Quarterly: Management Information Systems* 28, 1, 75–105.
- Hevner, Alan R. & Chatterjee, Samir 2010. *Design Science Research in Information Systems*. Boston, MA: Springer US.
- Hildenbrand, Tobias & Meyer, Johannes 2012. Intertwining Lean and Design Thinking: Software Product Development from Empathy to Shipment. In A. Maedche, hg. *Software for People - Fundamentals, Trends and Best Practices*. Springer Berlin Heidelberg.
- Keijzer-Broers, Wally J.W. & de Reuver, Mark 2016. Applying agile design sprint methods in action design research: Prototyping a health and wellbeing platform. In J. Parsons et al., hg. *Lecture Notes in Computer Science*, 9661 LNCS. Springer International.
- Lindberg, Tillmann & Meinel, Christoph & Wagner, Ralf 2011. Design Thinking: A Fruitful Concept for IT Development? *Journal of Consumer Research* 15, 2, 1–6.
- Luchs, Michael G. & Swan, Scott K. & Griffin, Abbie 2016. Design thinking : New product development essentials from the PDMA. M. Luchs & S. Swan & A. Griffin hg. John Wiley & Sons.
- Leimeister, Jan Marco 2014. *Collaboration Engineering*. Springer Berlin Heidelberg.
- March, Salvatore T. & Smith, Gerald F. 1995. Design and natural science research on information technology. *Decision Support Systems* 15, 4, 251–266.
- Österle, Hubert, Winter, Robert & Brenner, Walter 2010. *Gestaltungsorientierte Wirtschaftsinformatik: Ein Plädoyer für Rigor und Relevanz*. H. Österle, R. Winter, & W. Brenner, hg. Nürnberg: Infowerk.
- Peffers, Ken et al. 2008. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems* 24, 3, 45–77.
- Redlich, Beke & Rechten, Christopher & Schaub, Nina 2019. Auf das Mindset kommt es an! Design Thinking für industrie-nahe Dienstleistungen analog oder digital. *HMD Praxis der Wirtschaftsinformatik* 56, 1, 121-134
- Redlich, Beke et al. 2020. Wie Action Design Research und Design Thinking ein Innovationsprojekt zum Erfolg führen. In *WI2020 Zentrale Tracks*. GITO Verlag, 1616–1631.
- Robra-Bissantz, Susanne & Strahinger, Susanne 2020. *Wirtschaftsinformatik-Forschung für die Praxis*. *HMD Praxis der Wirtschaftsinformatik* 57, 2, 162–188.

- Schacht, Silvia, Morana, Stefan & Maedche, Alexander 2015. The Evolution of Design Principles Enabling Knowledge Reuse for Projects: An Action Design Research Project JOURNAL OF INFORMATION TECHNOLOGY THEORY AND APPLICATION The Evolution of Design Principles Enabling Knowledge Reuse for Projects: An Action Design Research Project.
- Scheele, Stephan & Mantwill, Frank 2019a. Gestaltung nutzerzentrierter Assistenzen im Produktdatenmanagement. ENTWERFEN ENTWICKELN ERLEBEN in Produktentwicklung und Design 2019
- Scheele, Stephan & Mantwill, Frank 2019b. Produktdaten-Apps zur Erweiterung der Lifecycle Produktivität H. P. Fröschle, hg. HMD Praxis der Wirtschaftsinformatik
- Sein, Maung K. et al. 2011. Action design research. MIS Quarterly: Management Information Systems 35, 1, 37–56.
- Simon, Herbert A. (Herbert Alexander) 1996. The sciences of the artificial. 3. ed., [N. Cambridge, Mass.: MIT Press.
- Urbach, Nils & Ahlemann, Frederik 2016. IT-Management im Zeitalter der Digitalisierung: Auf dem Weg zur IT-Organisation der Zukunft. Springer Berlin Heidelberg.
- Vaishnavi, Vijay & Kuechler, William; 2019. Design Science Research in Information Systems. <http://www.desrist.org/design-research-in-information-systems/>.
- Vetterli, Christophe et al. 2013. Jumpstarting Scrum with Design Thinking. Design Thinking Research: Building Innovators, 2, 2, 1–60.

Kontakt

Stephan Scheele
Daniel Mau
David Foullois
Prof. Dr.-Ing. Frank Mantwill

Helmut-Schmidt-Universität Hamburg
Institut für Maschinenelemente und Rechnergestützte Produktentwicklung
Holstenhofweg 85
22043 Hamburg