

Anforderungsmanagement in der Agilen Entwicklung Mechatronischer Systeme – ein Widerspruch in sich?

Nikola Bursac, Simon Rapp, Lukas Waldeier, Steffen Wagenmann, Albert Albers, Magnus Deiss, Volker Hettich

Anforderungen sind ein Kernbestandteil agiler Entwicklung und aufgrund der Charakteristika mechatronischer Systeme ist eine Bündelung von Anforderungen zur Realisierung von Entwicklungsgenerationen als Inkremente wirtschaftlich sinnvoll. Folglich ist ein Management der Anforderungen hinsichtlich ihrer Entwicklung, Verifizierung und Wiederverwendung notwendig. Anhand der Entwicklung von Werkzeugmaschinen werden die beobachteten Charakteristika der Entwicklung mechatronischer Systeme mit Einfluss auf Durchlaufzeit, Materialkosten und Aufwand beschrieben, diese sind: Fertigungsdokumentation, Bestellung, Lieferung, Hilfsmittel, Montage und Inbetriebnahme. Unter Berücksichtigung dieser Einflussfaktoren wird anschließend ein Vorgehen zur systematischen Wiederverwendung von Anforderungen bei der Entwicklung neuer Produkt- und Entwicklungsgenerationen von Werkzeugmaschinen vorgestellt. Dabei werden Qualitätsanforderungen, funktionale Anforderungen und Randbedingungen unterschieden. Für Qualitäts- und funktionale Anforderungen werden spezifische Vorgehensweisen im Rahmen der Realisierung einer Entwicklungsgeneration vorgeschlagen. Während die Erfüllung von Qualitätsanforderungen für eine Entwicklungsgeneration nach der Realisierung abzuprüfen ist, sind funktionale Anforderungen direkter Ausgangspunkt von Entwicklungsaktivitäten in einem Sprint. Randbedingungen werden als Begründung für Anforderungen nachvollziehbar und wiederverwendbar dokumentiert. Wesentliche Bestandteile des Vorgehens wurden in Jira implementiert und im Rahmen zweier Entwicklungsprojekte positiv evaluiert.

Keywords: Agilität, Scrum, Zielsysteme, PGE - Produktgenerationsentwicklung

Motivation

Insbesondere in der agilen Literatur ist häufig Kritik an starren Lasten- und Pflichtenheften zu finden. Dies ist darauf zurückzuführen, dass dieses Vorgehen gleichzeitig allen vier Kerngedanken des agilen Manifests widerspricht, wonach die Elemente auf der rechten Seite zwar wichtig, aber jene auf der linken wichtiger sind:

- „Individuen und Interaktionen mehr als Prozesse und Werkzeuge
- Funktionierende Software mehr als umfassende Dokumentation

- Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung
- Reagieren auf Veränderung mehr als das Befolgen eines Plans“
(Beck et al. 2001)

Hieraus entsteht häufig der Trugschluss, dass Anforderungen in agilen Vorgehensweisen unnötig sind. Nur weil die veraltete Ausführung von starren Anforderungen mittels Lasten- und Pflichtenheften seit Jahrzehnten nicht mehr angemessen ist (und im Übrigen in der Entwicklungspraxis auch nicht so gehandhabt wurde), folgt daraus nicht, dass Anforderungen als solche nicht benötigt werden.

Stand der Forschung

PGE – Produktgenerationsentwicklung

Die Entwicklung neuer Systeme wird durch das Modell der PGE – Produktgenerationsentwicklung beschrieben (Albers et al. 2015): Jede Systementwicklung erfolgt auf Basis eines Referenzsystems (Albers et al. 2019d). Dessen Bestandteile entstammen bereits existierenden oder in Entwicklung befindlichen Systemen sowie der zugehörigen Dokumentation. Referenzsystembestandteile können also beispielsweise Gestaltmodelle, Anforderungen oder Validierungsinformationen umfassen und unternehmenseigenen Vorgängergenerationen entstammen, anderen Produktlinien, Wettbewerbsprodukten oder Forschungsvorhaben. Abbildung 1 zeigt das Beispiel einer Stanzmaschine mit Teilen des zu Grunde liegenden Referenzsystems.

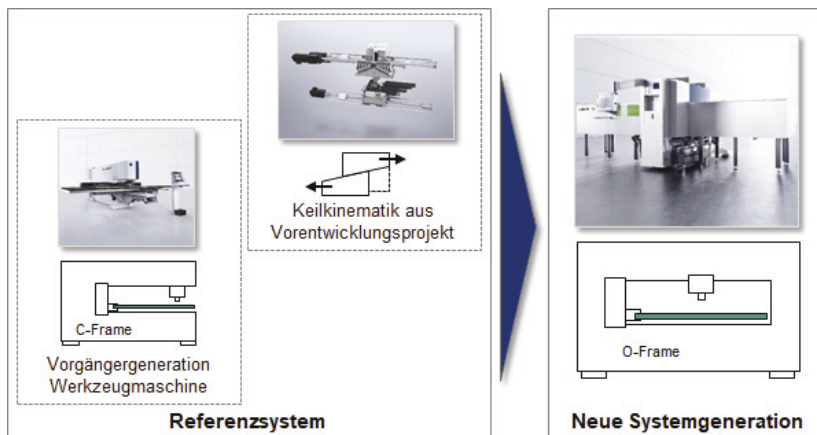


Abbildung 1: Stanzmaschine mit Teilen des zu Grunde liegenden Referenzsystems
(Albers et al. 2019d; Albers et al. 2019a)

Ausgehend vom Referenzsystem und dessen Subsystemen wird ein neues System durch die Kombination von Übernahme-, Gestalt- und Prinzipvariation von Subsystemen entwickelt (Albers et al. 2020). Referenzsysteminhalte werden so entweder direkt wiederverwendet oder dienen als Grundlage und Ausgangspunkt für weitere Entwicklungsaktivitäten. Dies kann am Beispiel der oben dargestellten neuen Systemgeneration einer Werkzeugmaschine veranschaulicht werden. Der Wechsel von C-Rahmen auf O-Rahmen stellt eine Prinzipvariation dar, während die Übernahme des Keilprinzips aus dem Vorentwicklungsprojekt eine Übernahmevariation darstellt. Die Stanzeinheit wurde hingegen mit einer Gestaltvariation weiterentwickelt. Charakteristika des Referenzsystems bzw. von dessen Subsystemen, beispielsweise deren Herkunft, als auch die verschiedenen Variationsarten beeinflussen das Entwicklungsrisiko (Albers et al. 2017). Ursächlich hierfür sind der jeweilige Zugang zu Wissen, der begrenzt sein kann, und technische Neuheit. So ist bei dem vorgestellten Beispiel des übernommenen Keilprinzips das Risiko deutlich höher, da das Referenzelement einem Vorentwicklungsprojekt entstammt, das zwar verifiziert ist, aber noch nicht im Markt eingeführt wurde. Aus den beabsichtigten Variationen und dem jeweils zu Grunde liegenden Referenzsystem mit seinen Charakteristika lassen sich unmittelbar erste Hinweise zur Prozessplanung in Form notwendiger Entwicklungsaktivitäten ableiten. So ist insbesondere bei Gestalt- und Prinzipvariationen mit Aktivitäten zur Ideenfindung, Modellierung von Prinzip und Gestalt sowie Validierung zu rechnen (Albers et al. 2016; Albers et al. 2019b). Das Modell der PGE lässt sich auch innerhalb der Entwicklung einer Produktgeneration auf verschiedene Entwicklungsgenerationen anwenden (Albers et al. 2017).

Agile Produktentwicklung

Ziel agiler Vorgehensweisen in der Produktentwicklung ist es, in komplexen Entwicklungskontexten schnell auf veränderte Anforderungen reagieren zu können und Entwicklungsrisiken durch die verhältnismäßig schnelle iterative Bereitstellung von Produktinkrementen zu begrenzen (Oestereich 2008). Eines der bekanntesten Frameworks für ein agiles Vorgehen ist Scrum (Schwaber und Sutherland 2007).

Bei Scrum werden Entwicklungszeiträume von maximal vier Wochen Dauer „Sprint“ genannt. Dabei ist es das Ziel bei der Softwareentwicklung in jedem Sprint ein Inkrement zu entwickeln. Anforderungen für die Entwicklung werden im „Backlog“ gesammelt. Als Grundlage eines Sprints werden entsprechend der Kapazitäten des Entwicklungsteams aus dem Backlog Anforderungen ausgewählt, die auf Grund des erwarteten Nutzenbeitrags am höchsten priorisiert sind. (Schwaber und Sutherland 2007).

Agile Entwicklungsweisen wie z.B. Scrum etablierten sich in der Software-Entwicklung. Auf Grund eingangs genannter Vorteile wird zunehmend auch eine Anwendung in der Entwicklung mechatronischer Systeme angestrebt (Atzberger et al. 2020). Diese Übertragung ist jedoch mit verschiedenen Herausforderungen verbunden, die durch spezifische Frameworks wie dem ASD – Agile Systems Design (Albers et al. 2019) adressiert werden. Beispielsweise sind Inkrementdauern wie bei Scrum von maximal vier Wochen bei mechatronischen Systemen derzeit mitunter schwer zu realisieren und wirtschaftlich nicht sinnvoll.

Ein wichtiger Schlüssel zur Bewältigung solcher Herausforderungen ist die systematische Nutzung von Referenzsystem beschrieben (Albers et al. 2019). Ausgehend vom Modell der PGE können Zwischenstufen in der Entwicklung eines neuen Systems, die auch als Produktinkremente aufgefasst werden können, als Entwicklungsgenerationen beschrieben werden (Albers et al. 2017). Darauf aufbauend kann für die Planung von Sprints das Wissen um Zusammenhänge zwischen Variationsarten und zur Realisierung voraussichtlich erforderlichen Aktivitäten der Produktentstehung genutzt werden (Wessels et al. 2019).

Anforderungsmanagement

Die Entwicklung eines technischen Systems wird durch das ZHO-Modell als Überführung eines Zielsystems (Anforderungen, Ziele, Randbedingungen) in ein Objektsystem (Dokumente, Artefakte) durch ein Handlungssystem (Entwickelnde, Ressourcen) beschrieben (Albers und Braun 2011; Bursac 2016; Lohmeyer 2013; Ebel 2015). Ebel (2015) beschreibt die Modellierung von Zielsystemen, die neben testbaren Anforderung zusätzliche Informationen beinhalten, wie die Begründung und Wechselwirkung zu anderen Zielsystemelementen. Die Anforderungen können hierbei in funktionale und nichtfunktionale Anforderungen unterschieden werden (Liu und Palen 2009; Birkhäuser et al. 2009). Funktionale Anforderungen bilden beschriebenes Verhalten ab, das ein technisches System erfüllen soll. Während nichtfunktionale Anforderungen unter anderem als Qualitätsanforderungen verstanden werden können. Funktionale Anforderungen werden dabei über den Entwicklungsprozess iterativ weiterentwickelt, priorisiert und verfeinert (Wynn et al. 2007).

Im Rahmen des vorliegenden Beitrags werden funktionale Anforderungen als Funktionsumfänge von Entwicklungsgenerationen verstanden und Qualitätsanforderungen als nicht funktionale Anforderungen, die nach der Fertigung einer Entwicklungsgeneration abgesichert werden müssen. Mit Rahmenbedingungen lassen sich funktionale

Anforderungen und Qualitätsanforderungen begründen, um auch über mehrere Produktgenerationen die Herkunft und Aktualität einer Anforderung nachvollziehen zu können.

Forschungsfragen und -Methoden

Obwohl umfangreiche Literatur zu den einzelnen Forschungsgebieten vorliegt, wurde bisher nicht betrachtet, inwiefern Anforderungen in einem agilen Vorgehen durch Entwicklungsgenerationen gebündelt werden sollten, um die Entwicklung, Verifikation und Wiederverwendung von Anforderungen zu ermöglichen. Im vorliegenden Beitrag wird daher das Forschungsziel verfolgt, Anforderungen in der agilen Entwicklung mechatronischer Systeme mittels Entwicklungsgenerationen nutzen und wiederverwenden zu können. Dazu werden folgende Forschungsfragen beantwortet:

1. Welche Charakteristika der agilen Entwicklung mechatronischer Systeme haben Einfluss auf das Anforderungsmanagement?
2. Wie sollte ein Anforderungsmanagement in der agilen Entwicklung mechatronischer Systeme unter Berücksichtigung dieser Charakteristika gestaltet sein?
3. Welcher Nutzen kann durch ein so gestaltetes Anforderungsmanagement in der agilen Entwicklung mechatronischer Systeme in der Entwicklungspraxis beobachtet werden?

Das Vorgehen richtet sich dabei nach der Design Research Methodology (DRM) von Blessing und Chakrabarti (2009): deskriptive Studie I zu Forschungsfrage 1; präskriptive Studie zu Forschungsfrage 2 und deskriptive Studie II zu Forschungsfrage 3.

Charakteristika der agilen Entwicklung mechatronischer Systeme und deren Einfluss auf das Anforderungsmanagement

Durch eine teilnehmende Beobachtung verschiedener agiler Entwicklungsprojekte von Werkzeugmaschinen und deren Automatisierungen konnten folgende Charakteristika beobachtet werden, die Einfluss auf die Releasezeiten von Entwicklungsgenerationen als Inkremente haben und dadurch zur Bündelung von funktionalen Anforderungen beitragen. Die Einflussfaktoren haben Auswirkungen auf Durchlaufzeit, Materialkosten und Aufwand.

- Ausgehend von einem Entwicklungsstand im 3D-CAD, wird ein Aufwand von etwa sechs Wochen für die Detaillierung, Erstellung, Prüfung und formale Freigabe von *Fertigungsdokumentation und Stücklisten* benötigt.

- Die *Bestellung* aller benötigten mechanischen und elektrischen Bauteile erfolgt in vielen Fällen stark automatisiert, jedoch ist bei neuen Bauteilen oder Änderungen ein entsprechender manueller Aufwand nötig.
- Die Durchlaufzeit bis zur Anlieferung variiert stark. Die Bandbreite geht zwischen der *Lieferung* lagerhaltiger Standard-Bauteile wie Schrauben und Sensoren von ein bis zwei Wochen hin zu Lieferzeiten umfangreicher Schweißbaugruppen, Gussteile oder bearbeiteter Strangpressprofile von über zwölf Wochen.
- Parallel zur Erstellung der physikalischen Bestandteile eines Produktes, werden auch *Fertigungs-, Transport und Montagehilfsmittel* benötigt, für die ähnliche Randbedingungen gelten.
- Nach Anlieferung der physikalischen Produktbestandteile folgen *Vormontagen, Montage, Verkabelung* sowie elektrische und softwaretechnische *Inbetriebnahme* von mindestens zwei bis vier Wochen.

Aus den Charakteristika ergibt sich, dass Releases von Entwicklungsgenerationen in der agilen Entwicklung mechatronischer Systeme eine längere Zeit benötigen. Diese ist dabei abhängig vom jeweiligen mechatronischen System. Insgesamt kann im Rahmen von Werkzeugmaschinen und deren Automatisierungen vom Beginn zur Auskopplung eines Inkrements bis zum Betrieb und Beginn der Qualifizierung von mindestens zwei bis vier Monaten ausgegangen werden. Hieraus ergeben sich wirtschaftlich sinnvolle Releasezeiten, die eine Balance zwischen Aufwand für die Erstellung der Entwicklungsgeneration und Wissenszugewinn durch die Iteration bilden. Eine zusätzliche Verkürzung kann unwirtschaftlich werden, da

- sich das Verhältnis zwischen eigentlicher Entwicklungsarbeit und den eher routinemäßigen Tätigkeiten zur Erstellung eines Inkrements verschlechtert,
- die Materialkosten stark ansteigen und
- die Komplexitätskosten steigen, da sich mehrere Stände gleichzeitig in verschiedenen Phasen der Erstellung und Qualifizierung befinden.

Eine Verkürzung der Releasezeiten von Entwicklungsgenerationen durch isolierte Inkremente einzelner Funktionen oder Baugruppen ist erstrebenswert. Auf diese Weise kann ein Spagat aus „potentially releaseable Inkrementen“ und tatsächlichen Releases gefunden werden. Allerdings ist dies sehr komplex, da sich Änderungen schnell auf Schnittstellen auswirken, welche dann starke Einschränkungen für das Inkrement zur

Folge haben. Beispielsweise gibt es bei einem sequentiellen Achsaufbau viele Abhängigkeiten hinsichtlich Dynamikanforderungen, Traglast und Bauraum zwischen den einzelnen Teilen des Systems. Eine weitere Möglichkeit zur Verkürzung der Releasezeiten von Entwicklungsgenerationen ist der Einsatz von Rapid Prototyping. Dies kann allerdings ebenfalls wirtschaftliche Nachteile mit sich bringen und erschwert die Qualifizierung der Serienfertigungsprozesse.

Einen weiteren Beitrag kann die systematische Wiederverwendung von Anforderungen leisten. Anforderungen, die sich beispielsweise aus den Umweltbedingungen am Aufstellungsort einer Maschine ergeben oder aus den gesetzlichen Vorgaben zur Produktsicherheit, bleiben häufig über mehrere Produktgenerationen gleich. Die genannten Beispiele können auf Basis der Literatur als Qualitätsanforderungen verstanden werden. Demgegenüber stehen funktionale Anforderungen, die stärker auf die Realisierung von Differenzierungsmerkmalen eines neuen Systems gegenüber bestehenden Systemen abzielen. Hier sind in verschiedenem Umfang Unterschiede für einzelne Produktgenerationen zu erwarten und dementsprechend andere Vorgehensweisen für die jeweils erforderlichen Variationen in der agilen Entwicklung.

Zur detaillierteren Untersuchung wurden Anforderungen an eine neue Produktgeneration analysiert, die in einem klassischen Lastenheft zu finden, sind. Dabei wird zunächst untersucht, welche Anforderungen im Hinblick auf ein agiles Vorgehen eher einer funktionalen Anforderung, einer Qualitätsanforderung oder einer Rahmenbedingung entsprechen (siehe Abbildung 2).

Aufgrund der relativ langen Dauer für die Entwicklung einer Entwicklungsgeneration, ist es entscheidend hoch priorisierte Anforderungen zu bündeln. Folglich sollte das Anforderungsmanagement ermöglichen:

1. dass Anforderungen wiederverwendet werden können,
2. dass Anforderungen iterativ und interdisziplinär weiterentwickelt werden können,
3. dass funktionale Anforderungen in Entwicklungsgenerationen gebündelt und validiert werden können,
4. dass Qualitätsanforderungen anhand der Entwicklungsgenerationen verifiziert werden können und
5. dass Rahmenbedingungen dokumentiert werden können, um auch nach mehreren Produktgenerationen nachvollziehen zu können, ob Anforderungen noch relevant sind.

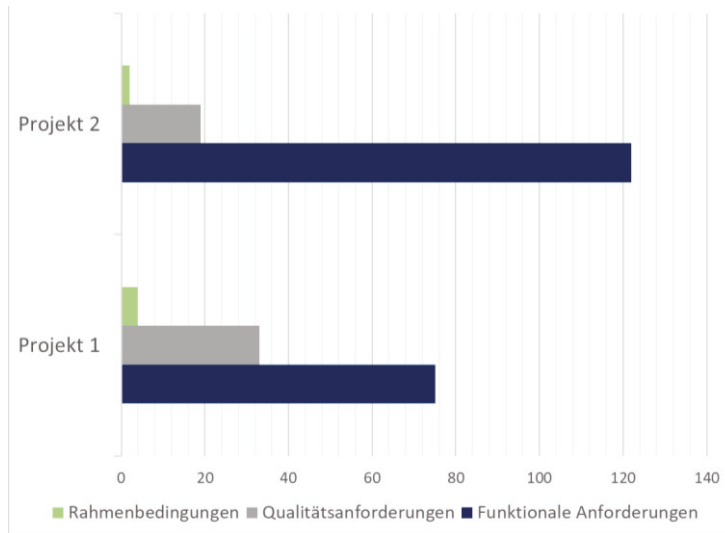


Abbildung 2: Aufteilung der Anforderungen

Auf diese Weise entsteht ein Anforderungsmanagement, das ein Wissensmanagementsystem darstellt. Durch die Dokumentation von Anforderungen und dazugehörigen Tests, wird transparent, welche Anforderungen im Rahmen welcher Entwicklungs-generation verifiziert wurden und welche Variationen in der aktuellen Entwicklungs-generation vorliegen.

Anforderungsmanagement in der agilen Entwicklung mechatronischer Systeme

Zur Realisierung der im vorherigen Kapitel abgeleiteten fünf Punkte, wurde das Vorgehen des Anforderungsmanagements weiterentwickelt. Für jede Entwicklungs-generation wird wie folgt vorgegangen:

- Funktionale Anforderungen werden priorisiert und im Rahmen von Entwicklungs-generationen heruntergebrochen (im agilen Kontext „refined“ genannt), entwickelt und verifiziert. Dabei wird zwischen Anforderungen unterschieden, die das Referenzsystem bereits erfüllt und Anforderungen, die sich gegenüber dem Referenzsystem verändern.

- Qualitätsanforderungen werden für jede Entwicklungsgeneration überprüft, ob durch die Prinzip- und Gestaltvariation Änderungen hervorgehen, die ein Refinement und eine Entwicklung erfordern, oder ob es genügt, beim Verifizieren zu überprüfen, ob die Qualitätsanforderung nach wie vorgegeben ist.
- Rahmenbedingungen werden kontinuierlich dokumentiert, damit die Anforderungen nachvollziehbar sind und priorisiert werden können.

The screenshot shows a JIRA issue titled 'Beispiel eee' with the following details:

- Details:** Type: Stakeholder Requirement, Status: TO REVIEW, Priority: nicht priorisiert, Version: PLA04_V01.03.00, Component: Keine, Keyword: Verbesserung, Version planning: Ja.
- Beschreibung:** Annotated with (5) Rahmenbedingungen dokumentieren.
- Anhänge:** Attachment 'eee.pptx' (53 kB) from 07.03.2021 10:03.
- Verknüpfungen:**
 - based on: VPLSTREQ-1642 Standardanforderung (Annotated with (1) Anforderungen wiederverwenden)
 - blocks: VPLFOVM-18126 Entwicklungsaufgabe (Annotated with (3) Entwicklungsaufgaben verlinken)
 - tested by: VPLFOVM-18125 Versuchsaufgabe (Annotated with (4) Versuchsaufgaben verlinken)
- Aktivität:** Comment by Bursac, Nikola (TW511bu) on 07.03.2021 09:59: 'Hallo Wagenmann, Steffen (TW513wa_st), kannst Du beantworten, ob ...' (Annotated with (2) Kollaborativ zusammenarbeiten).

Additional annotations on the 'Details' section:

- (3) Entwicklungsgeneration zuordnen (pointing to 'Stakeholder Requirement')
- (3) Veränderung gegenüber Referenzprodukt (pointing to 'nicht priorisiert')

Abbildung 3: Realisierung anhand eines beispielhaften JIRA Eintrags einer Anforderung

Für die Implementierung des Vorgehens in ein Tool, wurde das JIRA Plugin R4J (siehe Abbildung 3) gewählt, da JIRA in der Untersuchungsumgebung bereits in der agilen Entwicklung verwendet wurde. Die Entwickelnden sind daher bereits gewohnt, die JIRA

Einträge kontinuierlich weiterzuentwickeln. Sie nutzen beispielsweise die Möglichkeit Dokumente einzufügen (5), die Kommentarfunktion zur Zusammenarbeit (2) und die Verlinkung zu anderen Einträgen, wie beispielsweise Entwicklungs- und Verifizierungsaufgaben (3). Auf diese Weise konnte ein Konzept zur Wiederverwendung abgeleitet werden, indem Qualitätsanforderungen als Standardanforderungen definiert (1) und mit Standard Testcases verlinkt werden (4). Die Anforderungen können untereinander mit Hilfe der Rang Funktion priorisiert werden und in Versionen den Entwicklungsgenerationen zugeordnet werden (3).

Beobachteter Nutzen des entwickelten Anforderungsmanagements

Das vorgeschlagene Vorgehen wurde zunächst mit Hilfe von Expertenbefragungen evaluiert und anschließend in zwei Entwicklungsprojekten pilotiert und evaluiert.

Für die Expertenbefragung in Abbildung 4 wurden n=9 Experten um ihre Einschätzung hinsichtlich folgender Aussagen gebeten.




Aussage	Wertung	Anzahl befragte Mitarbeiter aus Fachbereichen (n=9)	
Aussage 1: Steigerung der Effektivität pro Fachbereich Der vorgestellte Ansatz steigert die Effektivität beim operativen Arbeiten, d.h. beim Anlegen und Bearbeiten von Anforderungen / Tests, im Vergleich zur Arbeit mit Excel.	 0 5 4	Produktmanagement	3
Aussage 2: Verwaltung der Daten Die Dokumentationsform unterstützt die einfache und übersichtliche Verwaltung von Product Requirements, z.B. die Anpassung der Struktur oder das Einfügen / Entfernen von Requirements / Tests in die Struktur.	 0 0 9	Entwicklung	2
Aussage 3: Aufwand / Nutzenverhältnis angemessen pro Fachbereich Der zusätzliche Dokumentationsaufwand im Vergleich zu Excel ist im Verhältnis zum Nutzen/Mehrwert des Ansatzes angemessen.	 0 1 8	Qualifizierung	3
		Qualitätsmanagement	1

Abbildung 4: Einschätzung von Experten hinsichtlich der Aussagen.
Stimme zu (grün), stimme teilweise zu (gelb), stimme nicht zu (rot)

Die durchgeführten Expertenbewertungen unterstützen diese Einschätzung. Die Integration in den Alltag fordert jedoch einen hohen initialen Aufwand (Tool Auswahl und Implementierung, Schulungen, KVP, Begleitung in den Projekten), der nach Einschätzung der Experten durch die Verbesserungen im Bereich Nachvollziehbarkeit, Kommunikation, Effektivität und Wiederverwendung gerechtfertigt ist.

Die Anwendung des Vorgehens wurde darüber hinaus in zwei Entwicklungsprojekten evaluiert. Das erste Entwicklungsprojekt begann vor 1,5 Jahren und das zweite vor 6 Monaten. Anhand der Änderungshistorie und der JIRA Einträge wird nachvollzogen, inwiefern die Weiterentwicklung des Anforderungsmanagements von einem dokumentenbasierten Vorgehen hin zu einem modellbasierten Vorgehen sich verändert hat.

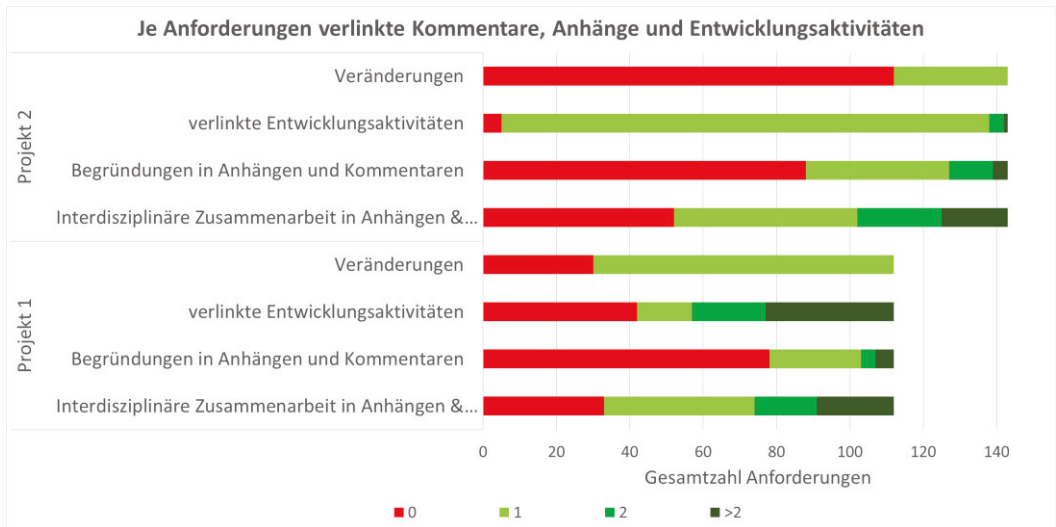


Abbildung 5: Zu Anforderungen verlinkte Kommentare, Anhänge und Entwicklungsaktivitäten in Entwicklungsprojekten

Wie aus Abbildung 5 ersichtlich ist, wurde durch das Nutzen der Kommentarfunktion eine Förderung der interdisziplinären Zusammenarbeit erzielt. Dies ermöglicht es Anforderungen iterativ und interdisziplinär weiterzuentwickeln (2). Insbesondere kritische Anforderungen weisen vermehrt Kommentare mit Begründungen auf, die häufig Gegenstand von Diskussionen sind (5). Durch die Dokumentation der Rahmenbedingungen kann die Relevanz von umgesetzten Anforderungen in späteren Entwicklungs-generationen geprüft werden (5). Komplementierend hierzu erlaubt die Verlinkung von bearbeiteten Entwicklungsaufgaben einen direkten Bezug auf bereits durchgeführte Aktivitäten (1, 2).

Diskussion und Ausblick

Im Kontext zunehmender agiler Ansätze im Maschinen- und Anlagenbau, stellt sich die Frage, ob Anforderungen noch einen Platz haben, oder nicht eher zu veralteten Entwicklungsmethoden gehören. Befasst man sich jedoch tiefer mit der Materie, stellt man zurecht fest, dass Anforderungen in agilen Frameworks wie Scrum einen noch höheren Stellenwert haben als bisher (Schwaber und Sutherland 2007). Diese werden im Backlog regelmäßig gesammelt, aktualisiert und priorisiert. Agil zu arbeiten heißt folglich nicht, weniger favorisierte Arbeitsschritte in der Entwicklung wegzulassen, sondern eine regelmäßige und auch späte Veränderung von Anforderungen zu fördern. So wie es auch in der Zielsystemforschung (Muschik 2011; Lindemann 2009) seit vielen Jahren gefordert wird. In der agilen Entwicklung mechatronischer Systeme stellt sich also nicht die Frage, ob Anforderungen benötigt werden, sondern wie diese verwaltet werden können. Dabei ist die Arbeit mit Anforderungen effizient, wenn sie wiederverwendet und für Entwicklungsgenerationen gebündelt werden können.

Limitationen der vorliegenden Ergebnisse entstehen, bei der Zuordnung der Kommentare bzgl. der interdisziplinären Zusammenarbeit und der Begründungen, da diese nicht immer trennscharf klassifiziert werden können. Die Wiederverwendung von Anforderungen (1) ist implementiert, kann jedoch erst bei folgenden Produktgenerationen evaluiert werden, da zunächst eine Basis erstellt werden muss. Folglich konnte hier keine Evaluation durchgeführt werden. Scherer et al. (2017) zeigen allerdings, dass hierbei hohe Potentiale der Wiederverwendung von Anforderungen bestehen.

In zukünftigen Projekten wird auf Basis der Priorisierung funktionaler Anforderungen erfolgskritische funktionale Anforderungen datengetrieben validiert. Stark kostentreibende funktionale Anforderungen werden dadurch bereits frühzeitig durch die gezielte Analyse von Kundennutzungsdaten der Referenzsysteme validiert und auf ihre Notwendigkeit geprüft. Erstellte Analysen werden standardmäßig in JIRA verlinkt und dadurch die tatsächliche Nutzung von Funktionen in den Referenzsystemen aufgezeigt. Die Analysen sind so gestaltet, dass sie sich bei Veränderungen der Datengrundlage automatisch anpassen. Sie sind demnach auch für spätere Produktgenerationen sinnvoll nutzbar. Dies leistet einen nachhaltigen Beitrag im Sinne der Begründung und der Aktualität von Anforderungen.

Danksagung

Der vorliegende Beitrag entstand im Rahmen des Forschungsprojekts MoSys – Menschorientierte Gestaltung komplexer System of Systems. MoSys wird durch das

Bundesministerium für Bildung und Forschung (BMBF) im Programm „Innovationen für die Produktion, Dienstleistung und Arbeit von morgen“ gefördert und vom Projektträger Karlsruhe (PTKA) betreut.

Literaturverzeichnis

- Albers, A.; Rapp, S.; Birk, C.; Bursac, N. (2017): Die Frühe Phase der PGE - Produktgenerationsentwicklung. In: SSP 2017.
- Albers, A.; Rapp, S.; Fahl, J.; Hirschter, T.; Revfi, S.; Schulz, M. et al. (2020): Proposing a Generalized Description of Variants in Different Types of Systems by the Model of PGE. In: DESIGN 2020.
- Albers, A.; Rapp, S.; Spadinger, M.; Richter, T.; Birk, C.; Marthaler, M. et al. (2019a): Das Referenzsystem im Modell der PGE – Produktgenerationsentwicklung: Vorschlag einer generalisierten Beschreibung von Referenzprodukten und ihrer Wechselbeziehungen.
- Albers, Albert; Braun, Andreas (2011): A generalised framework to compass and to support complex product engineering processes. In: IJPD 15.
- Albers, A.; Bursac, N.; Rapp, S. (2017): PGE – Produktgenerationsentwicklung am Beispiel des Zweimassenschwungrads. In: Forschung Ingenieurwesen 81.
- Albers, A.; Bursac, N.; Wintergerst, E. (2015): Produktgenerationsentwicklung – Bedeutung und Herausforderungen aus einer entwicklungsmethodischen Perspektive. In: SSP 2015.
- Albers, A.; Heimicke, J.; Spadinger, M.; Reiss, N.; Breitschuh, J.; Richter, T.; Bursac, N. et al. (2019b): A systematic approach to situation-adequate mechatronic system development by ASD - Agile Systems Design. In: Procedia CIRP.
- Albers, A.; Rapp, S.; Peglow, N.; Stürmlinger, T.; Heimicke, J.; Wattenberg, F.; Wessels, H. (2019c): Variations as Activity Patterns: A Basis for Project Planning in PGE – Product Generation Engineering. In: Procedia CIRP 84.
- Albers, A.; Rapp, S.; Spadinger, M.; Richter, T.; Birk, C.; Marthaler, F. et al. (2019d): The Reference System in the Model of PGE: Proposing a Generalized Description of Reference Products and their Interrelations. In: ICED 2019.
- Albers, A.; Reiss, N.; Bursac, N.; Richter, T. (2016): iPeM – Integrated Product Engineering Model in Context of Product Generation Engineering. In: CIRP 50.
- Atzberger, A.; Nicklas, S.; Schrof, J.; Weiss, S.; Paetzold, K. (2020): Agile Entwicklung physischer Produkte.
- Beck, K.; Beedle, M., van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J. et al. (2001): Manifest für Agile Softwareentwicklung.
- Birkhäuser, B.; Pottebaum, J.; Koch, R. (2009): Unterstützung von Einsatzentscheidungen der Feuerwehr auf Basis IT-unterstützter Kräftekoordination. In: Informatik 2009.
- Blessing, L.; Chakrabarti, A. (2009): DRM: A Design Reseach Methodology. London: Springer.
- Bursac, N. (2016): Model Based Systems Engineering zur Unterstützung der Baukastenentwicklung im Kontext der Frühen Phase der Produktgenerationsentwicklung. Karlsruhe.
- Ebel, B. (2015): Modellierung von Zielsystemen in der interdisziplinären Produktentstehung. Karlsruhe.

Lindemann, U. (2009): Methodische Entwicklung technischer Produkte. Berlin: Springer.

Liu, S.; Palen, L. (2009): Spatiotemporal mashups: A survey of current tools to inform next generation crisis support. In: IS-CRAM 2009.

Lohmeyer, Q. (2013): Menschzentrierte Modellierung von Produktentstehungssystemen unter besonderer Berücksichtigung der Synthese und Analyse dynamischer Zielsysteme. Karlsruhe.

Muschik, S. (2011): Development of Systems of Objectives in Early Product Engineering. Karlsruhe.

Oestereich, B. (2008): Agiles Projektmanagement. Erfolgreiches Timeboxing für IT-Projekte. In: HMD 45 (2).

Scherer, H.; Albers, A.; Bursac, N. (2017): Model Based Requirements Engineering for the Development of Modular Kits. In: Procedia CIRP 60.

Schwaber, K.; Sutherland, J. (2007): The Scrum Guide.

Wessels, H.; Heimicke, J.; Rapp, S.; Graubeger, P.; Richter, T.; Matthiesen, S.; Albers, A. (2019): Sprint planning in mechatronic system development based on reference system elements. In: 17. KT. Aachen.

Wynn, D.; Eckert, C.; Clarkson, P. (2007): Modelling iteration in engineering design. In: ICED.

Kontakt

Dr.-Ing. Nikola Bursac
Lukas Waldeier, M. Sc.
Steffen Wagenmann, B. Eng.
Dr.-Ing. Volker Hettich
TRUMPF Werkzeugmaschinen
Johann-Maus-Straße 2
71254 Ditzingen
www.trumpf.com

Prof. Dr.-Ing. Albert Albers
Simon Rapp, M. Sc.
Karlsruher Institut für Technologie (KIT)
IPEK – Institut für Produktentwicklung
Kaiserstraße 10
76131 Karlsruhe
www.ipek.kit.edu

Magnus Deiss
ENDEKO GmbH
Rippoldsauerstr. 23
70372 Stuttgart
www.endeko.de