

Software-Defined Product Features

Reduktion physischer Varianz durch Virtualisierung

Adrian Barwasser,
Joachim Lentes,
Oliver Riedel und
Nikolas Zimmermann, Stuttgart

Kundenansprüche zwingen viele produzierende Unternehmen dazu individualisierte Produkte anzubieten, um wettbewerbsfähig zu bleiben. Dies hat weitreichende und kostspielige Auswirkungen auf nahezu jeden Bereich des Produktlebenszyklus von der Entwicklung der Produkte, über die Beschaffung und Lagerung von Material und Einzelteilen, bis hin zur Produktion. Software-Defined Product Features (SDPF) ist ein Ansatz der Produktentwicklung, welcher Möglichkeiten schafft, bisher physikalisch ausgeführte Produkteigenschaften und -funktionen mittels Software zu realisieren und dadurch Komplexität und Kosten ganzer Produktfamilien zu reduzieren. Im folgenden Beitrag wird der Begriff „Software-Defined Product Features“ definiert und in den weiteren Kontext des SDx (Software-Defined Anything) und insbesondere des SDM (Software-Defined Manufacturing) gesetzt. Anschließend wird der Stand der Forschung bezüglich einer neuen Methode vorgestellt, welche eine Potenzialanalyse für den Einsatz von SDPF an verschiedenen Produkten ermöglichen und Handlungsempfehlungen für deren Weiterentwicklung liefern soll.*)

Ausgangssituation und Motivation

Der Markt vieler Investitions- und Gebrauchsgüter wird von den vielfältigen Ansprüchen potentieller Kunden dominiert. Dieser Umstand zwingt produzierende Unternehmen dazu, individualisierte Produkte anzubieten, um wettbewerbsfähig zu bleiben. In der Folge steigt der physische Komplexitätsgrad und die damit verbundenen Herausforderungen nehmen immer weiter zu. Als Beispiel hierfür eignet sich die Betrachtung der Ausstattungsvarianten eines PKW-Modells. Götz beziffert die Anzahl der Modellvarianten, welche im Jahr 2007 Teil der BMW Group waren (BMW, Mini und Rolls Royce) auf ca. 350. Bei bis zu 500 Sonderausstattungen ergeben sich so rund 10^{31} Ausstattungsvarianten. Andere Marken und Modelle folgen diesem Trend mit VW Golf – 10^{23} , Opel Astra – 10^{17} , Ford Focus – 10^{16} [1].

Eine derartig große äußere Komplexität beeinflusst nahezu jeden Bereich des Produktlebenszyklus. Piller beschreibt die Auswirkungen dieser Komplexität auf die Fertigung und angrenzende Bereiche im Detail [2]. Die Auswirkungen reichen von der Entwicklung der Produkte, über die Beschaffung und Lagerung von Material und Einzelteilen, bis hin zur Produktion.

In der Beschaffung von Material und Einzelteilen muss aufgrund der Produktindividualisierung eine größere Anzahl verschiedener Komponenten in kleineren Mengen beschafft werden, was tendenziell die Verhandlungsposition des Einkaufs schwächt und zu höheren Einkaufspreisen führt. Aufgrund der Schwierigkeit, den genauen Bedarf der Kunden an einzelnen individualisierten Produkten vorherzusagen, müssen die verschiedenen Komponenten entweder in ausreichend großer Zahl vorgehalten werden, um einen ununterbrochenen Produktionszyklus sicherstellen zu können. Dies hat zur Folge, dass mehr Platz für die Lagerung von Komponenten vorgesehen werden muss. Auch die Komplexität des Logistiksystems kann in der Folge zunehmen. In der Entwicklung steigt der zeitliche Aufwand durch die Notwendigkeit, mehr Varianten zu entwickeln, zu

verwalten und zu aktualisieren, sobald sich Änderungen ergeben [3]. Die Produktionsplanung wird durch die geringeren Losgrößen und den häufigeren Produktionswechsel komplexer [4]. In der Produktion steigen neben den Rüstkosten auch die Kosten für eventuell zusätzlich benötigte Produktionsmaschinen und diversifizierte Produktionslinien [5]. Letztlich macht sich auch in Vertrieb und Kundenservice das erweiterte Produktspektrum durch erhöhte Komplexität bemerkbar. Eine hohe physische Komplexität und Varianz der herzustellenden Produkte führen also zu erhöhten Kosten für das betroffene Unternehmen [5].

Stand und Defizite in Forschung und Praxis

Unternehmen sind sich der zuvor dargestellten Herausforderungen bewusst und versuchen über Varianten- und Komplexitätsmanagement die Auswirkungen und Ursachen dieser Problemstellung zu adressieren. Beispielsweise werden die negativen Auswirkungen auf Beschaffung und Logistik durch Ansätze wie Just-in-Time und Just-in-Sequence gemindert [6]. Paradigmen wie Modularität in Entwicklung und Produktion haben längst Einzug in den Entwicklungspro-

*) Hinweis

Bei diesem Beitrag handelt es sich um einen von den Mitgliedern des ZWF-Advisory Board wissenschaftlich begutachteten Fachaufsatz (Peer-Review).

zess gehalten und ermöglichen den Ingenieuren, bereits früh im Produktentwicklungsprozess den Anstieg an Komplexität und Kosten einzudämmen [7]. Im Falle der Modularität geschieht dies dadurch, dass Produkte aus Modulen unterschiedlichen Umfangs aufgebaut werden, welche über Schnittstellen miteinander verbunden sind. Dadurch ist es dem Hersteller möglich, neue Produkte und Varianten existierender Produkte durch den Austausch einzelner Module zu generieren, ohne aufwendige Änderungen an den gleichbleibenden Modulen vornehmen zu müssen [8]. Dies erübrigt kostenintensive Neuentwicklungen und ermöglicht die Nutzung von Skaleneffekten in Beschaffung und Produktion. Außerdem kann das Unternehmen auf existierende Produktionsabläufe und -erfahrung zurückgreifen. Sofern dieser Ansatz genutzt wird, um eine große Anzahl kundenspezifischer Endprodukte mit möglichst geringem Aufwand anbieten zu können, wird auch von Mass Customization gesprochen [9]. Insgesamt ist der Trend der Modularität in der Betrachtung des Produktentstehungsprozesses dadurch gekennzeichnet, dass die Variantenbildung möglichst spät im Produktentwicklungsprozess stattfindet und so die innere Komplexität des Unternehmens (gekennzeichnet z. B. durch die Anzahl verschiedener herzustellender Komponenten, Fertigungsabläufe) geringgehalten werden kann, während die äußere Komplexität (Anzahl angebotener Produkte aus Sicht des Kunden) so groß wie notwendig wird. All diese Ansätze haben gemein, dass sie dazu verleiten, bestehende Konstruktionen auf einen kleinsten gemeinsamen Nenner zu reduzieren, die darin enthaltenen Annahmen jedoch nicht weiter zu hinterfragen. Daraus leitet sich der Forschungsbedarf nach einer Methode ab, welche letzteres befördert.

Software-Defined Product Features

Ein Ansatz zur Beherrschung der Auswirkungen und Ursachen von Komplexität, welcher seit mindestens zehn Jahren in Einzelfällen zum Einsatz kommt, aber noch nicht formal beschrieben und in den regulären Produktentwicklungsablauf integriert wurde, ist der Ansatz der Software-Defined Product Features (SDPF). Bei SDPF handelt es sich um einen Ansatz, bisher physikalisch definierte Produkteigenschaften und -funktionen mittels Software abzubilden und dadurch Komplexität und Kosten des betroffenen

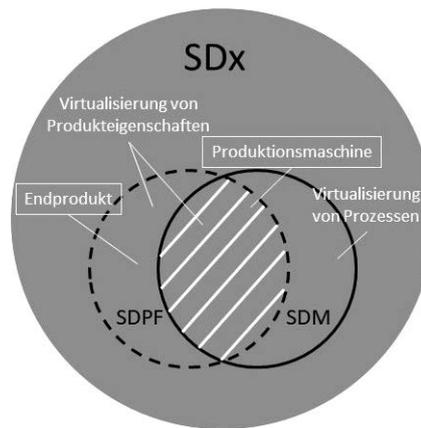


Bild 1. Übersicht SDx: Gemeinsamkeiten und Unterschiede zwischen SDPF und SDM

Produkts oder ganzer Produktfamilien zu reduzieren. Auf der Grundlage per Software definierter Produkteigenschaften und -funktionen und damit auch Varianten kann dann beispielsweise ein Customize-to-Order als Ergänzung zu Build-to-Forecast und Build-to-Order realisiert werden [10]. Damit setzt sich dieser Ansatz grundsätzlich von zuvor erwähnten Methoden ab, da die Virtualisierung der Produkteigenschaften bisher noch nicht betrachtete Optionen offenlegt.

In der Forschung geht der Begriff „Software-Defined“ bis in die 1970er-Jahre zurück und fand zunächst hauptsächlich Anwendung im Bereich der Elektronik, vor allem in Form von Software-Defined Networking und Software-Defined Radio [11]. In den letzten zehn Jahren ist der Begriff durch den Fortschritt der betroffenen Technologie verstärkt in den Fokus der Forschung zur Produktentstehung gekommen. Dies hat dazu geführt, dass unter dem Überbegriff Software-Defined Anything (SDx, manchmal auch als Software-Defined Everything bekannt) an verschiedenen Ansätzen geforscht wird, anhand derer sich Produkte, Prozesse und Dienstleistungen mittels Software verbessern lassen. Von besonderem Interesse ist hierbei der Ansatz des Software-Defined Manufacturing (SDM).

Software-Defined Manufacturing

SDM beschäftigt sich mit der Automatisierung und letztendlich Autonomisierung von Produktionslinien und beschreibt den Prozess der Ableitung der Herstellungsbeschreibung auf der Grundlage einer von der Fertigungstechnik unabhängigen Produktbeschreibung, die Geometrie, Material, Qualität und Funktion des Produkts bestimmt. Die

Idee besteht darin, dass die Herstellung einzelner Produkte von den technischen Merkmalen der Produktionsanlagen losgelöst ist und stattdessen teilweise oder vollständig über Software definiert wird [12]. Grundvoraussetzung für SDM ist eine vollständige, maschinenlesbare Produktbeschreibung, auf deren Basis kurz vor Produktionsbeginn die benötigten Geräte sowie Fertigungsmaschinen und -prozesse automatisch per Software definiert und parametrisiert werden können, damit ein bestimmtes Produkt hergestellt werden kann. Darüber hinaus lassen sich aus dieser Beschreibung Transport-, Logistik- und Produktionsschritte ableiten [12]. Für die Betrachtung von SDPF ist das Konzept des SDM insofern von Interesse, als dass SDM in großen Teilen als ein Sonderfall von SDPF gesehen werden kann (Bild 1): Das Produkt, dessen Produkteigenschaften dabei von Software definiert werden, ist eine Produktionsmaschine bis hin zu einer ganzen Produktionseinrichtung, welche mehrere Produktionsmaschinen samt Infrastruktur beinhaltet. Ein signifikanter Unterschied zwischen den beiden Konzepten besteht darin, dass SDM neben der Produktdefinition auch den Produktionsprozess betrachtet.

Aufgrund der Überschneidungen zwischen SDM und SDPF lassen sich Fortschritte in einem Bereich der Forschung zumindest in Teilen auf den anderen übertragen. Bisher gibt es jedoch auch für SDM kein formalisiertes methodisches Vorgehen, welches zur Virtualisierung von Produkteigenschaften führen würde.

Beispiele für Software-Defined Product Features

Automobilhersteller ersetzen beispielsweise Motorsysteme verschiedener Leistungsstärken durch ein einziges, physisch identisches System. Über Softwareeinstellungen wird dann festgelegt, wie viel Prozent der eigentlich möglichen Leistung je Variante abgerufen werden kann (Bild 2).

Durch die Virtualisierung von Varianten durch Software und Parametrisierung wird die physische Komplexität der Varianten drastisch reduziert, da eine Anpassung der Software spät im PEP geschehen kann und vergleichsweise geringe Auswirkungen auf den restlichen Entstehungsprozess hat. Diesem Vorteil steht der Nachteil gegenüber, dass aus physischer Sicht überdimensionierte Bauteile verwendet werden – z. B. Zylinder, wel-

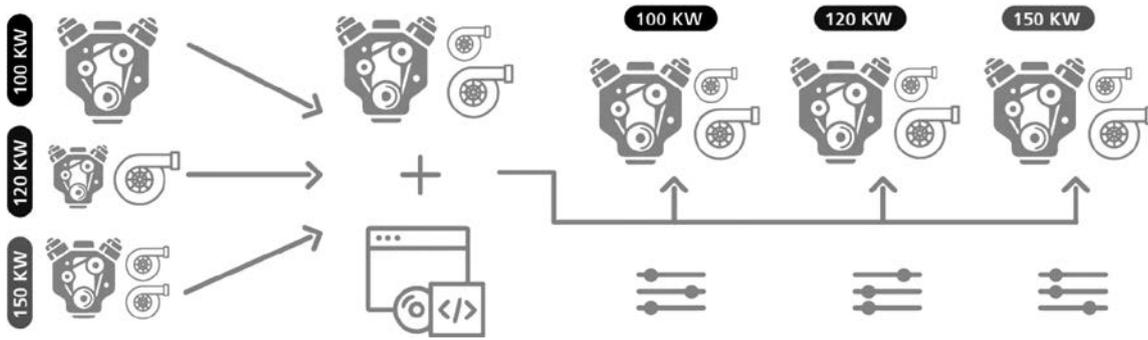


Bild 2. Reduktion physischer Varianten durch Virtualisierung einer Produkteigenschaft (hier: Leistung)

che den Belastungen eines 150 kW-Motors standhalten könnten und entsprechend ausgelegt sind, nun aber für einen 100 kW-Motor verwendet werden. Dadurch gehen Einspar- und Leichtbaupotenzial verloren. Ob die Vor- oder Nachteile überwiegen und eine signifikante Kostensenkung erzielt werden kann, muss daher von Fall zu Fall berechnet werden. Dies kann unter Umständen recht kompliziert sein, besonders dann, wenn eine ganzheitliche Abschätzung der Einsparungen durch Reduktion der Komplexität oft schwer zu quantifizieren ist. Außerdem gibt es bisher noch keine Beschreibung eines methodischen Vorgehens für den Einsatz Software-definierter Produkteigenschaften und die Produktentwickler können sich daher bestenfalls an dem Beispiel einzelner erfolgreicher Umsetzungen orientieren.

Das Beispiel der virtualisierten Leistungsvariantenbildung im Motorenbau zeigt einen Sonderfall, bei welchem innerhalb von Produktfamilien mehrere Varianten zusammengefasst werden, indem die qualitativ hochwertige Hardware mit Softwareapplikationen in ihrer Funktionalität stufenweise parametrisiert werden kann. Dies ist jedoch nicht die einzige Art, wie SDPF ein Produkt beeinflussen kann. Ein weiteres Beispiel Software-definierter Produkteigenschaften findet sich in der Adaptronik – oft auch Smart Structures oder Smart Materials genannt. Hier werden gezielt Aktoren (häufig Piezofolienaktoren) eingesetzt, um strukturmechanische Anforderungen zu erfüllen.

Das Smartphone ist ein Beispiel aus dem täglichen Gebrauch, welches sich in der Wahrnehmung des Nutzers nahezu vollständig über Software-definierte Produkteigenschaften definiert. Das darauf installierte Betriebssystem sowie die individualisierte Konfiguration via Apps machen einen Großteil des Nutzererlebnisses aus. Zudem entfällt durch den Touchscreen eine große Zahl mechanischer Teil-

le, deren Eingabefunktion und Design nun durch Software definiert werden können.

Diese Beispiele zeigen, dass vielfältige Kategorien von Produkteigenschaften bei der Entwicklung eines methodischen Vorgehens für den Einsatz Software-definierter Produkteigenschaften in Betracht gezogen werden müssen.

Entwicklung einer Methode für den Einsatz von SDPF

Aus dem vorangegangenen Kapitel wird ersichtlich, dass zwar bereits etablierte Ansätze zur Beherrschung von Variantenvielfalt und Komplexität existieren, diese aber das Potenzial, welches durch die Virtualisierung von Produkteigenschaften und -funktionen entsteht, noch nicht vollständig ausschöpfen. Eine neue Methode soll diese Lücke schließen. Bei der Entwicklung dieser Methode können grundsätzlich zwei Ansätze verfolgt werden: die Schaffung komplett neuer Prozesse und Methoden oder die Anpassung existierender an die neu definierten Anforderungen. Der Vorteil einer komplett neuen Methode ist die kreative Freiheit, welche innovative Lösungen begünstigt. Der Vorteil einer Anpassung existierender Ansätze liegt im reduzierten Aufwand und vor allem in der Senkung der Einstiegshürden für neue Anwender. Die Einführung einer neuen

Methode stellt für Anwender immer eine Hürde dar – sowohl wirtschaftlicher, als auch psychologischer Natur. Sofern diese Methode jedoch auf bereits bekannten und bewährten Ansätzen basiert, gestaltet sich die Einführung naturgemäß einfacher. Aus diesem Grund ist es zielführend, auf Ansätze aufzubauen, welche bereits weitestgehend etabliert sind und diese dann den neuen Bedürfnissen anzupassen. Dies betrifft Prozesse und Methoden verschiedener Granularitätsstufen. Auf einer übergeordneten Ebene kann zum Beispiel an das Vorgehen nach VDI 2221 [13] angeknüpft werden, da dieses in der deutschen Industrie weite Verbreitung findet [14]. Gleichzeitig müssen gängige Methoden, wie z. B. der Morphologische Kasten, Quality Function Deployment (QFD), an das neue Ziel der Integration Software-definierter Produkteigenschaften angepasst werden. Auch die Kombination von Methoden der klassischen Konstruktion mit Methoden der Softwaretechnik ist denkbar. Das Ergebnis ist eine Variante der bisher gängigen Prozesse und Methoden, welche den Anwender bei der Entwicklung von Produkten mit Software-definierten Produkteigenschaften unterstützt und Transparenz bezüglich der Vor- und Nachteile gegenüber herkömmlichen Produkten auf der Grundlage physischer Variantenbildung schafft.

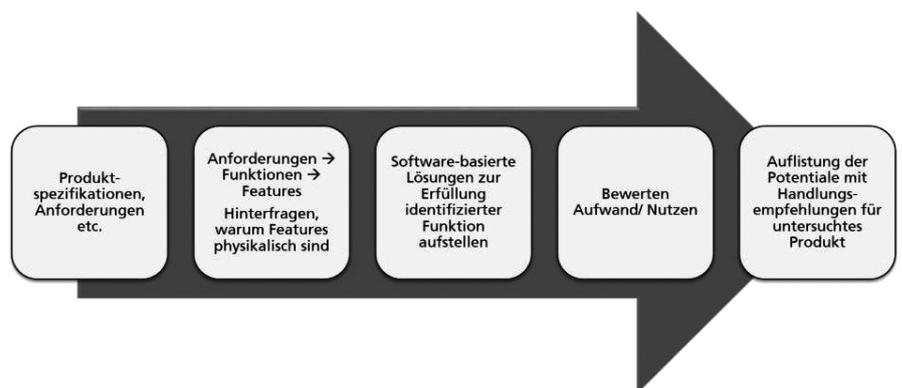


Bild 3. Grobstruktur der angestrebten Methode

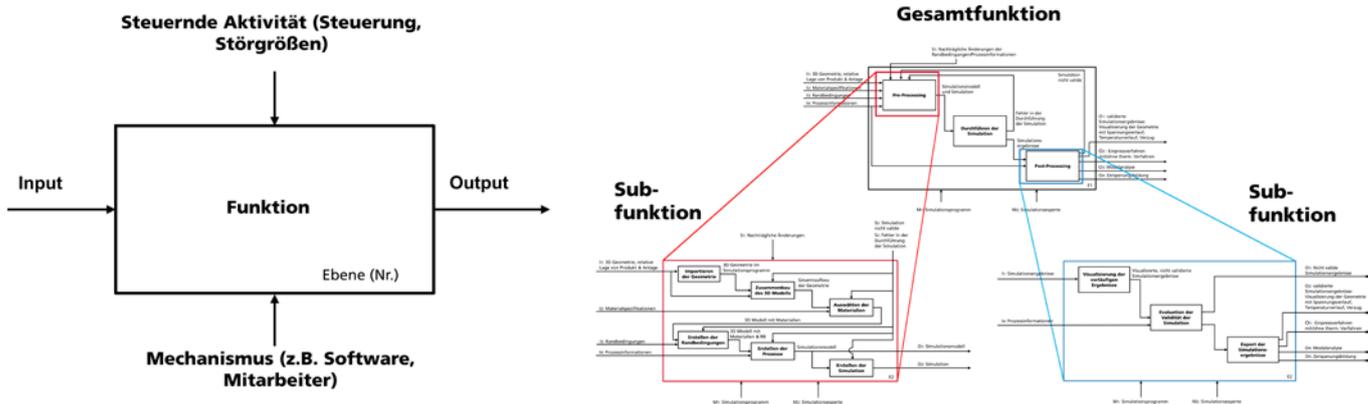


Bild 4. Funktionsweise SADT (Structured Analysis and Design Technique)

Es stellt sich daher die Frage, wie die zuvor erwähnten Methoden der Produktentwicklung abgewandelt werden müssen, um das Potenzial von Software-definierten Produkteigenschaften auszuschöpfen. Dabei liegt der Fokus – im Gegensatz zu kompletten Neuentwicklungen – zunächst auf der Weiterentwicklung existierender Produkte.

Im Rahmen des laufenden Projekts „Cognitive Engineering“ am Fraunhofer IAO wurde bereits der Kern der angestrebten Methodik entwickelt. Bild 3 zeigt die Grobstruktur der angestrebten Methode von Produktspezifikationen des Ausgangsprodukts bis hin zu den gesuchten Potenzialen für den Einsatz von SDPF, inklusive Handlungsempfehlungen und Aufwandseinschätzung.

Der Kern der Methodik besteht in der Analyse der Anforderungen, Funktionen und vorhandenen physikalischen Features (Produkteigenschaften). Dazu muss der bestehende Zusammenhang zwischen Produktfeatures und deren Funktion analysiert und aufgebrochen werden, um ein alternatives – Software-basiertes – Feature als neue Lösung einbringen zu können. Hierfür müssen geeignete Methoden ausgewählt werden, um der natürlichen Voreingenommenheit für die bereits umgesetzte Lösung entgegenzuwirken und die Kreativität der Produktentwickler zu fördern. Beispielsweise lässt sich mittels der SADT-Methode [15] (alternativ davon abgeleitet IDEFO) eine Produktfunktion schrittweise in immer feinere Subfunktionen herunterbrechen (Bild 4).

Die Aufgliederung der Beziehungen zwischen den einzelnen Subfunktionen in Input, Output, steuernde Aktivität und Mechanismus befördert ein tiefergehendes Verständnis des Produkts. Insbesondere die Angriffspunkte der Mechanismen zeigen existierende sowie fehlende

Software-definierte Aspekte des betrachteten Produkts und können dem Nutzer Anhaltspunkte für die Weiterentwicklung liefern. [15]

Bei der Betrachtung möglicher Änderungen ist die Nutzung existierender Steuergeräte von besonderem Interesse, da dadurch geringere Eingriffe in das Produkt und damit geringere Aufwände erwartet werden können. Nicht jede auf diese Weise gefundene Änderungsmaßnahme führt jedoch zu einem besseren Produkt. Für potentiell vielversprechende Maßnahmen muss eine Aufwand-/Nutzen-Analyse durchgeführt werden. Sofern ein günstiges Verhältnis von Aufwand zu Nutzen besteht, werden für die identifizierten Maßnahmen entsprechende Handlungsempfehlungen entworfen, welche die Weiterentwicklung des Produkts in die richtige Richtung weisen. Diese ergeben sich größtenteils aus den zuvor neu entwickelten Verknüpfungen aus Funktionen und Software-basierten Features.

Zusammenfassung und Ausblick

Am Fraunhofer IAO wird derzeit eine Methode entwickelt, welche Unternehmen dabei unterstützen soll, die Herstellkosten ihrer Produkte zu reduzieren, Flexibilität besonders in der Produktion zu erhöhen und Innovationen zu befördern. Dies wird erreicht, indem bisher physikalisch ausgeführte Produkteigenschaften mittels Software realisiert werden.

Die anstehenden Forschungsarbeiten lassen sich grob in drei Arbeitsbereiche unterteilen. Zum einen muss eine passende Produktbeschreibung definiert werden, welche sich dafür eignet, die Anforderungslisten, Produktspezifikationen und weitere Dokumente verschie-

denster Produkte und Unternehmen in eine standardisierte Form zu bringen, welche als Grundlage für die weiteren Schritte dienen kann. Dabei wurde eine Beschreibung basierend auf SysML als vielversprechender Kandidat identifiziert. Zum anderen müssen weitere geeignete Methoden für die einzelnen Schritte des Gesamtprozesses identifiziert und evaluiert werden. Wichtige Faktoren sind die Anpassungsfähigkeit an die neuen Bedürfnisse sowie die Verbreitung in der Wirtschaft. Letztlich muss die Wirksamkeit der angestrebten Methode über iterative Anwendungsbeispiele validiert und verbessert werden. Dazu werden nach Abschluss initialer Tests bei Fraunhofer IAO auch industrielle Pilotanwender gesucht. Es wird angestrebt, die Resultate eines solchen Praxistest in einer späteren Ausgabe des ZWF zu veröffentlichen.

Literatur

1. Götz, A.: Zukunftsstandort Deutschland? Automobilproduktion (2007) 2, S. 16–19
2. Piller, F.: Mass Customization. Ein wettbewerbsstrategisches Konzept im Informationszeitalter. Gabler Edition Wissenschaft Markt- und Unternehmensentwicklung, Dt. Univ.-Verl., Wiesbaden 2006, S. 178–198
3. Meentken, F.: Entwicklung eines Vorgehensmodells zur Einführung eines globalen Qualitätsmanagementsystems. Zugl.: Dissertation, IWF der TU Berlin; Berichte aus dem Produktionstechnischen Zentrum Berlin, Faunhofer Verlag, Stuttgart 2016
4. Olbrich, R.; Battenfeld, D.: Variantenvielfalt und Komplexität – kostenorientierte vs. marktorientierte Sicht. Journal für Marketing 44 (2005) 3/4, S. 161–173
DOI: 10.1007/BF03032078
5. Bititci, U.S.; Carrie, A.S.: Strategic Management of the Manufacturing Value Chain. In: Proceedings of the International Conference of the Manufacturing Value-Chain August '98, (IFIP - The International Federation for In-

formation Processing, 2). Springer-Verlag, Troon, Scotland, UK. Boston, MA 1998
DOI: 10.1007/978-0-387-35321-0

6. Jansen, H.: Lean Production in der mittelständischen Industrie. Springer-Verlag, Berlin, Heidelberg 1993, S. 45f
7. Picot, A.; Baumann, O.: Modularität in der verteilten Entwicklung komplexer Systeme. Chancen, Grenzen, Implikationen. Journal für Betriebswirtschaft 57 (2007) 3/4, S. 221–246
DOI: 10.1007/s11301-007-0029-8
8. Baldwin, C. Y.; Clark, K. B.: Design Rules. Volume 1: The Power of Modularity. MIT Press, Cambridge, Mass 2000
DOI: 10.7551/mitpress/2366.001.0001
9. Piller, F.; Kumar, A.: For Each, their Own. The Strategic Imperative of Mass Customization. Industrial Engineer 38 (2006) 9, S. 40–46
10. Dangelmaier, W.; Lentjes, J.; Döring, A.: Neue Produktionsstrategien für die Automobilindustrie. Automobiltechnische Zeitschrift ATZ-Produktion 2 (2009) 2, S. 52–55
11. Clifford, C.; Landuyt, R.; Schmitt, H.: Microprocessor Based, Software Defined Television Controller. IEEE Trans. Consumer Electron. CE-24 (1978) 3, S. 436–442
DOI: 10.1109/TCE.1978.267047
12. Lechler, A.; Riedel, O.; Coupek, D.: Virtual Representation of Physical Objects for Software Defined Manufacturing. 24th International Conference on Production Research (ICPR 2017)
DOI: 10.12783/dtetr/icpr2017/17652
13. Verein Deutscher Ingenieure (Hrsg.): VDI 2221: Design of Technical Products and Systems – Model of Product Design. VDI, Düsseldorf 2019
14. Hentschel, C.: Konzeptionelle Produktentwicklung als Impulsgeber für radikale Innovation – TRIZ in der Konstruktionssystema-

tik des VDI. In: Khare, A.; Kessler, D.; Wirsam, J.: Marktorientiertes Produkt- und Produktionsmanagement in digitalen Umwelten. Springer-Gabler-Verlag, Wiesbaden 2018, S. 75–84
DOI: 10.1007/978-3-658-21637-5_6

15. Ross, D. T.: Structured Analysis (SA): A Language for Communicating Ideas. IEEE Transactions on Software Engineering, Band SE-3 (1977) 1, S. 16–18
DOI: 10.1109/TSE.1977.229900

■ Die Autoren dieses Beitrags

Adrian Barwasser, M. Sc., geb. 1991, studierte an der TU Darmstadt Maschinenbau. Nach Abschluss seines Studiums in 2017 arbeitete er zunächst für den Ingenieurdienstleister Engineering People. Seit Januar 2018 arbeitet er als Wissenschaftlicher Mitarbeiter im Team für Digital Engineering am Fraunhofer IAO. Sein Schwerpunkt liegt auf der Optimierung von Engineering- und Produktionsprozessen mit digitalen Werkzeugen und innovativen Methoden.

Dipl.-Ing. Joachim Lentjes, geb. 1969, studierte Maschinenwesen an der Uni Stuttgart und arbeitet seit dem als Wissenschaftler am Fraunhofer IAO, wo er im Jahr 2009 die Position des Leiters Digital Engineering übernahm. Zudem ist er Lehrbeauftragter an der Universität Stuttgart. Seine Forschungsarbeiten befassen sich insbesondere mit der Produktentwicklung und Produktion der Zukunft mit innovativen Strategien, Methoden und Werkzeugen.

Prof. Dr. Oliver Riedel, geb. 1965, studierte Technische Kybernetik an der Universität Stuttgart und promovierte dort an der Fakultät der Konstruktions- und Fertigungstechnik. Nach seiner Tätigkeit am Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO) war er bei der CENIT AG Systemhaus und der Silicon-Graphics Inc. im Bereich Professional Services für das Themenfeld Virtuelle Produktentstehung in leitender Funktion tätig. Danach war Herr Prof. Riedel für die Prozessintegration und das Informationsmanagement im Produktprozess bei der AUDI AG verantwortlich. 2010 wechselte er als Leiter Informationstechnologie und Prozessintegration Produktprozess innerhalb des VW Konzerns. Ab 2012 war er bei der

AUDI AG verantwortlich für die Steuerung der Planungsprozesse und die Koordination produktionsrelevanter IT. Seit November 2016 ist Prof. Riedel Institutsleiter des Instituts für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW) an der Universität Stuttgart, Inhaber des Lehrstuhls Produktionstechnische Informationstechnologien und seit Juni 2018 Institutsleiter des Fraunhofer-Instituts für Arbeitswirtschaft und Organisation (IAO).

Dipl.-Ing. Nikolas Zimmermann, geb. 1983, studierte Maschinenbau am KIT und ist seit 2011 Wissenschaftlicher Mitarbeiter am Fraunhofer IAO. Seine Forschung befasst sich hauptsächlich mit Prozessoptimierung, Produktionsplanung, Additive Manufacturing und IT-basierter Engineering-Unterstützung. Darüber hinaus verantwortet er das Digital Engineering Lab, ein Demonstrationszentrum für Forschungsaktivitäten und Demonstratoren in Stuttgart, Deutschland.

■ Summary

Software-Defined Product Features – Reduction of Physical Variance through Virtualization. Customer demands force many manufacturing companies to offer individualized products in order to remain competitive. This has far-reaching and costly effects on almost every area of the product life cycle, from the development of the product, the procurement and storage of materials and individual parts, to manufacture of the products. Software-Defined Product Features (SDPF) is a product development approach that creates opportunities to realize previously physically executed product features and functions using software, thereby reducing the complexity and costs of entire product families. The following article defines the term „Software-Defined Product Features“ and puts it into the broader context of SDx (Software-Defined Anything) and especially SDM (Software-Defined Manufacturing). Subsequently, the state of research regarding a new method is presented, which should enable a potential analysis for the use of SDPF on different products and provide recommendations for action for their further development.

Bibliography
DOI 10.3139/104.112433
ZWF 115 (2020) 11; page 824–828
© Carl Hanser Verlag GmbH & Co. KG
ISSN 0947–0085