Risiken von Technical Debt identifizieren und managen

Thomas Aicher, Huan Dong und Birgit Vogel-Heuser
Lehrstuhl für Automatisierung und Informationssysteme, TU München

Oft muss bei der Behebung eines technischen Problems unter Zeitdruck zwischen einer aufwandsarmen „suboptimalen“ Lösung und der aufwändigeren „besseren“ Lösung entschieden werden. Diese Frage stellen sich Ingenieure, Techniker und Entscheider häufig bei der Entwicklung, dem Betrieb bzw. der Inbetriebnahme eines Embedded Systems (ES). Dabei müssen diese die Projektsituation, die verbleibende Laufzeit und Kosten berücksichtigen. Unter dem Begriff der technischen Schuld, also der Technical Debt (TD), wird die bewusste Entscheidung gegen eine bessere, sinnvollere Lösung verstanden. Obwohl TD im klassischen Software-Engineering ein bekanntes und erforschtes Phänomen ist, wird es bis dato in den Bereichen der Embedded Systems und Mechatronik, wie beispielsweise ein Produktionssystem, noch wenig beachtet und ist in Folge dessen wenig beherrscht. Die daraus resultierenden finanziellen Schäden für Unternehmen sind erheblich. Aus diesem Grund ist die Bereitstellung transparenter Grundlagen für bewusste Entscheidungen zur Inkaufnahme von TD von besonderem Interesse. Anhand einer Erweiterung der TD-Klassifikation aus dem Software Engineering und einer domänenübergreifenden Fallstudie wird TD und ihre Auswirkungen sowie das Potential zur Beherrschung von TD diskutiert. Um die geforderte Transparenz zu erreichen, werden für Embedded Systems sowohl die kurzfristigen als auch die langfristigen Auswirkungen aufgezeigt und gegenübergestellt.

Einleitung und Motivation

Der Begriff „Technical Debt“ (TD) wurde im Bereich des Software Engineerings vor mehr als zwei Jahrzehnten eingeführt [1]. TD beschreibt die langfristigen negativen Effekte, die entstehen, wenn kurzfristig bewusst eine suboptimale Lösung anstatt einer aufwändigeren „optimalen“ Lösung gewählt wird. TD gilt im Software Engineering als ein bekanntes Forschungsthema, konnte aktuell aber noch keinen Einzug im Bereich der Embedded Systems (ES) und Mechatronik, wie beispielsweise ein Produktionssystem, finden. Ein häufiges Einsatzfeld von ES sind Maschinen und Anlagen, aber auch mehr oder weniger komplexe mechatronische Systeme wie Züge, Fahrzeuge oder Baumaschinen. Alle diese mechatronischen Systeme können als eine Komposition aus mechanischen, elektrischen und Software-Komponenten betrachtet werden, die in den verschiedenen Lebenszyklusphasen – Entwickelung, Inbetriebnahme und Wartung (vgl. Abb. 1) – zusammenwirken [2]. Insbesondere die Inbetriebnahme, Wartung und weitere Optimierung stellen zusätzliche Phasen dar in denen TD auftreten kann (Abb. 1). Abhängig von der Größe und Komplexität der Maschine und des zu produzierenden Produktes wird sowohl im Werk als auch auf der Baustelle getestet. Häufig kann ein ES in einer Maschine vor der finalen Inbetriebnahme nicht vollständig getestet werden. Deshalb stehen die Inbetriebnehmer oft unter hohem Zeitdruck und es müssen Fehler kurzfristig auf der Baustelle behoben bzw. neue Lösungen entwickelt

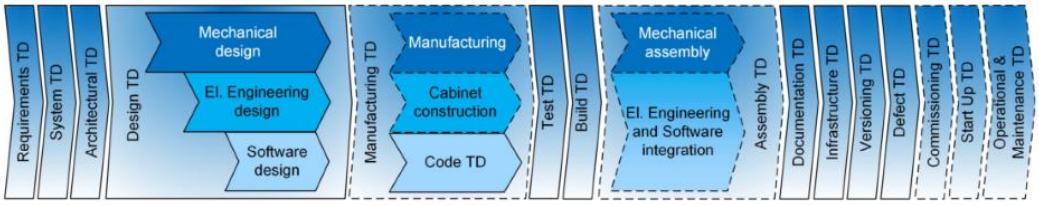

Abb. 1: ES in der Automatisierung für den Maschinen -und Anlagenbau [2]

werden. In dieser Inbetriebnahmephase sind viele Beispiele für TD zu finden. Die späteren Wartungskosten solcher Bug-Fixes oder on-site Entwicklungen häufig parallel zu systematischeren Lösungen aus der Konstruktion überschreiten die zuvor eingesparten Kosten bei der Inbetriebnahme. Die hohen Wartungs- und Pflegekosten resultieren aus den langen Lebenszykluszeiten (häufig über 20 Jahre) von Maschinen und Anlagen sowie den häufigen über 100 parallel betreibenden Maschinen. Ziel der Kostenbetrachtung im Rahmen von TD ist es nicht nur Herstellkosten oder Inbetriebnahmekosten, sondern auch die Betriebskosten für den Kunden zu betrachten.

Für mechatronische Systeme [3] ist jedoch nicht nur die Software sondern auch die Disziplinen Mechanik und Elektrik/Elektronik zu berücksichtigen. Im Folgenden wird TD zwar immer noch aus der Sicht der Disziplin Software betrachtet, aber in Wechselwirkung zu den anderen Disziplinen. Anhand der Fallstudie eines Maschinen- und Anlagenherstellers, wird die TD-Klassifikation erweitert und eine Visualisierung von TD für ES in diesem Bereich vorgeschlagen.

Vorgehensweise und Fallbeispiel

In Interviews mit Maschinen- und Anlagenherstellern konnten wir verschiedene typische Beispiele für TD in ES sammeln. Parallel dazu wurde mittels einer web-basierten Online-Umfrage eine repräsentative Studie mit mehr als 80 Unternehmen aus verschiedenen Anwendungsbereichen von ES durchgeführt. Die aus dem Software Engineering bekannten TD-Klassen könnten damit um multidisziplinäre und Lebenszyklus-spezifische Klassen für ES in der Mechatronik erweitert werden. Im Folgenden wird dazu eine solche TD-Klasse für die Fallstudie eines weltweit führenden Maschinen- und Anlagenherstellers vorgestellt, der unter anderem Roboter in seine Anlagen integriert.

Während der Inbetriebnahme einer Anlage, haben Techniker eine vertauschte Verdrahtung innerhalb eines Roboters festgestellt (Fehler in der Elektrik/Elektronik). Obwohl die Techniker das Verdrahtungsproblem am Roboter hätten korrigieren können, erfolgte dies nicht, sondern die Softwareschnittstelle der Steuerungssoftware wurde angepasst. Diese Änderung wurde weder in der Softwaredokumentation noch im Elektroschaltplan dokumentiert. Hierdurch ist eine TD in der Dokumentation und eine TD in der Disziplin Elektrotechnik während der Inbetriebnahme entstanden. Beides ist durchaus typisch, weil sich Software häufig einfacher und schneller ändern lässt.

Nach einigen Jahren wird der Roboter durch einen neuen gleichen Roboter mit korrekter Verdrahtung ausgetauscht. Da die Software jedoch noch immer für die Steuerung des falsch verdrahteten Roboters ausgelegt war, führte der neue Roboter unerwartete (invertierte) Manöver aus und verursachte dabei Schäden an der Anlage.

Ein Teil der Anlage musste wegen diesem Vorfalls repariert werden. Dieser Schaden verursachte für den Maschinenlieferanten und den Betreiber einen beträchtlichen finanziellen Schaden.

Da die Dokumentation der Verdrahtung und Steuerungssoftware des Roboters nach dem Auffinden des Fehlers und der Anpassung innerhalb der Software nicht angepasst wurde, liegen drei TD-Klassen vor (vgl. Abb. 2) erstens: Commissioning TD, weil die falsche Verdrahtung innerhalb des Roboters bei der Integration des Roboters in die Anlage im Werk entstand, dort aber aufgrund eines fehlenden Tests nicht identifiziert wurde. Erst während der Inbetriebnahme auf der Baustelle wurde der Fehler identifiziert und aus Zeitgründen die Lösung (bug fix) in der Software realisiert.

Die zweite TD, Code TD, ist durch die Überarbeitung der Software-Schnittstellen entgegen der Standardschnittstelle entstanden, die zur Notwendigkeit der Pflege von verschiedenen Versionen dieser Schnittstelle führt. Die dritte TD ist die Dokumentations-Schuld, die durch die fehlende Dokumentation entsteht, so dass beim Austausch des Roboters durch einen neuen kein Hinweis auf die notwendige Anpassung der falsch ausgeführten Schnittstelle vorliegt. Der entstandene Schaden beruht auf diesen drei TD-Arten.

Erweiterte TD-Klassen für ES in der Mechatronik und im Maschinen- und Anlagenbau

Die diskutierten TDs für ES im Maschinen- und Anlagenbau erweitern die Klassifikationen des Software Engineering [6] (siehe Abb. 2). Die zusätzlichen Dimensionen beziehen sich auf die weiteren Phasen des Lebenszyklus von solchen ES wie beispielsweise Systemintegration, Inbetriebnahme sowie Betrieb und Wartung. Zudem sind für betroffene Disziplinen d.h. Mechanik und/oder Elektrik/Elektronik, auch weitere zusätzliche Dimensionen angegeben [2].

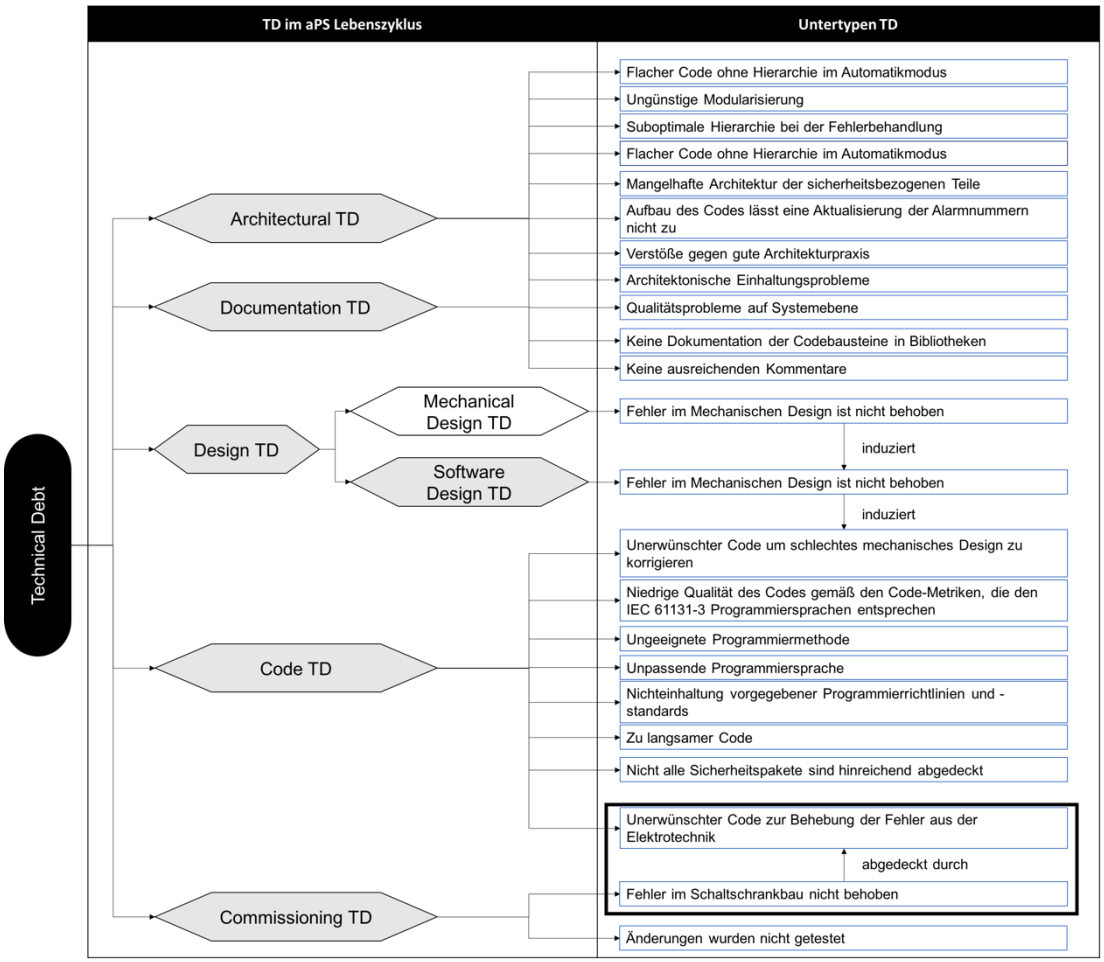

Abb. 1: Eine Erweiterung der TD-Klassifikation für die ES in der Mechatronik insbesondere im Maschinen- und Anlagenbau: Graue Felder repräsentieren den vorhandenen Typ laut Li et al. [6] und weiße Felder repräsentieren neue Untertypen laut [4] und [5] sowie die Fallstudie in diesem Beitrag (zwei Untertypen im gepunkteten Linienrechteck)

Kosten von TD in ES für den Maschinen- und Anlagenbau (Ergebnisse der Online Umfrage)

Aufbauend auf verschiedenen fragebogenbasierten Untersuchungen zur Reife von Software-Entwicklung und zum Engineering von ES im Maschinen- und Anlagenbau [7] im aktuellsten Fragebogen (aktiv geschaltet seit Juli 2017) wurden sechs Fragen zu TD eingebunden. Die beteiligten Unternehmen sind in verschiedenen Branchen tätig, darunter Automotive, Medizin, Robotik, etc. Die Teilnehmer stammen sowohl aus dem Bereich Software, aber auch aus den Disziplinen Elektrik/Elektronik und Mechanik. Insgesamt liegen bereits 80 ausgefüllte Fragebögen vor, bisher konnten jedoch nur 51 berücksichtigt werden. Nach Ende des Fragebogens im November 2017 können die restlichen eingegangen Daten einbezogen werden. Im Folgenden werden einige ausgewählte Aspekte vorgestellt:

TD verursacht im Bereich der Software am häufigsten zusätzliche und langfristige Aufwände (siehe Abb. 3).

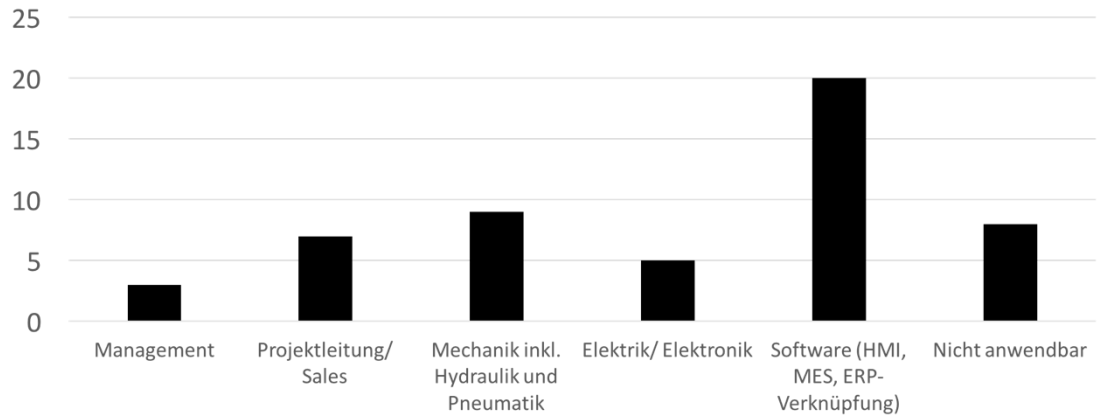

Abb. 3: Langfristige Aufwände durch TD in den verschiedenen Disziplinen

Bei der Betrachtung der Aufwände werden zwei Fälle unterschieden: Erstens der kurzfristige Nutzen aufgrund der häufig aufwandsärmeren suboptimalen Lösung und zweitens der zusätzliche langfristige Aufwand aufgrund der suboptimalen Lösung (Abb. 4). Dabei verursacht TD durchschnittlich erhebliche außerplanmäßige Aufwände, die die kurzfristigen Vorteile übersteigen.

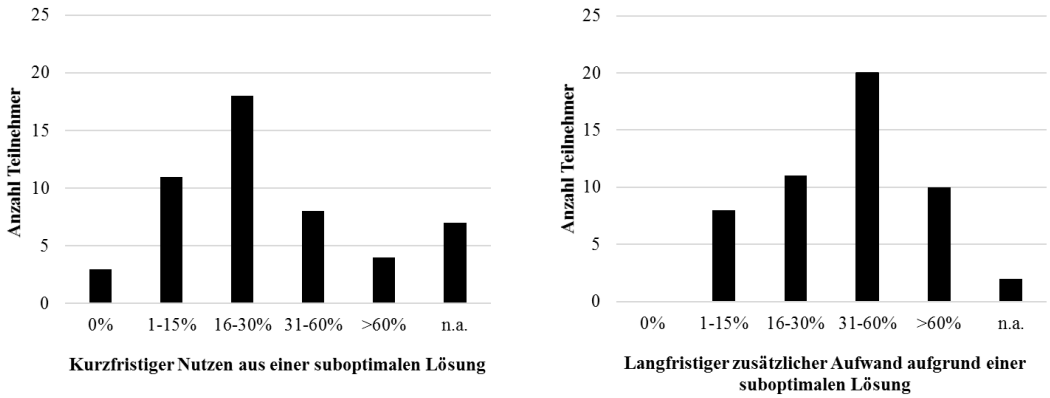

Abb. 4: Kurzfristiger Nutzen aus der suboptimalen Lösung (18 Teilnehmer wählten 16-30 % Nutzen) im Vergleich zu langfristigen zusätzlichen Aufwand (20 Teilnehmer wählten 31-60 % Aufwand)

Visualisierung zur Frühwarnung von TD und Kostenabschätzung

Um die Risiken hoher späterer Kosten von TD frühzeitig zu erkennen und die Entscheidung TD in Kauf zu nehmen, bewusster und unter Kenntnis der voraussichtlichen Kosten zu treffen, ist diese zu visualisieren (Abb. 5). Insbesondere bei TD zwischen verschiedenen Disziplinen oder verschiedenen Lebenszyklusphasen erlaubt die Visualisierung eine höhere Transparenz über Kostenstellen hinweg. Mittels einer Ampelfunktion wird die Kritikalität der TD angezeigt. Am Fallbeispiel des falsch verdrahteten Roboters werden durch die Visualisierung die zusätzlichen Kosten für die Disziplin Software für eine oder mehrere Anlagen verdeutlicht (siehe Abb. 5). Nach Abschluss der Inbetriebnahme einer Anlage, können bewusst gewählte TD nur mit hohen Aufwand wieder durch die aufwändigere optimale Lösung ersetzt werden.

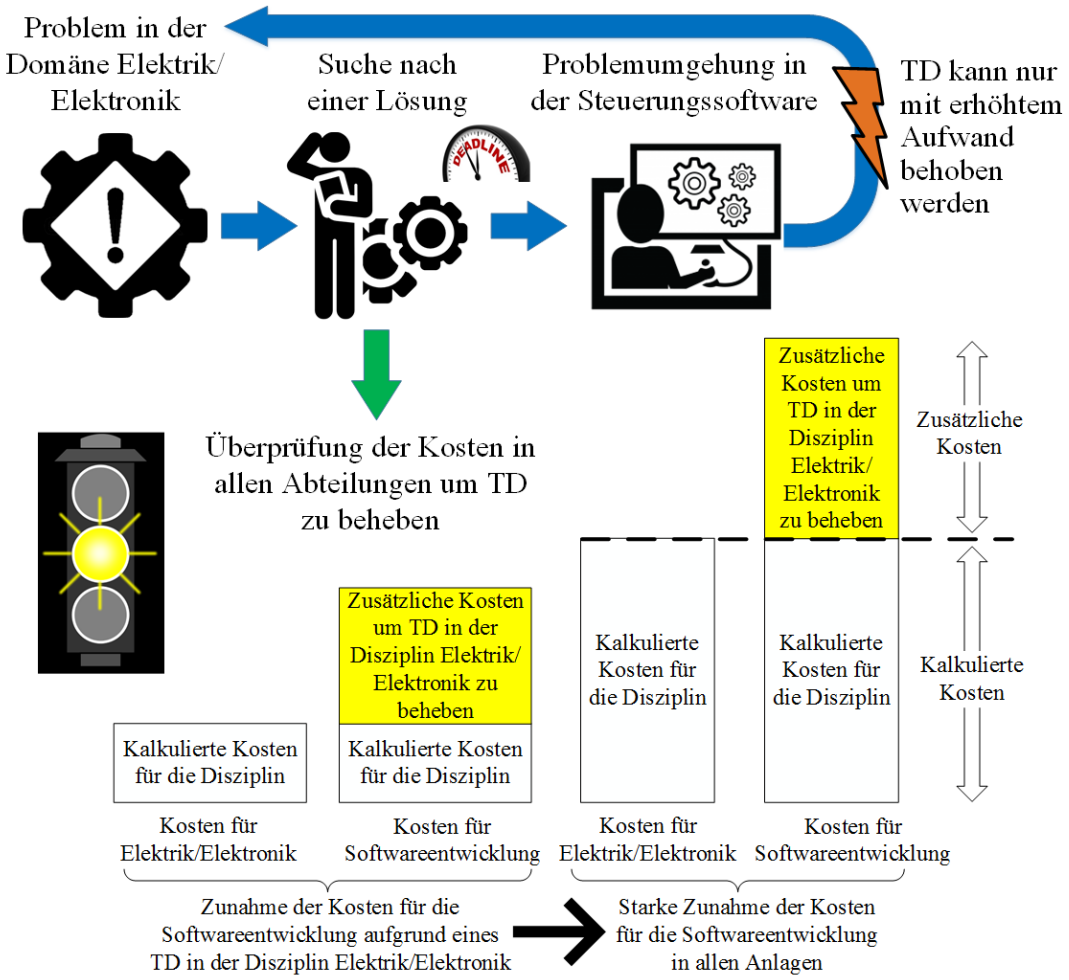

Abb. 5: Überprüfen der Kosten in den verschiedenen Disziplinen und Projekten zur Frühwarnung von TD

Um solche Kostenabschätzungen visualisieren zu können, ist die TD in die Nachkalkulation sukzessive einzubinden und damit die Datenbasis iterative zu verbessern.

Schlussfolgerung und Ausblick

In dem klassischen Software-Engineering ist TD ein bekanntes Phänomen, welches jedoch aufgrund der zusätzlichen Disziplinen Mechanik und/oder Elektrik/Elektronik, nicht ohne weiteres für das Software-Engineering von ES und mechatronischen Systemen insbesondere des Maschinen– und Anlagenbaus überführt werden kann. Die Auswirkungen von TD über die Disziplingrenzen hinaus gilt es ebenfalls in zukünftigen Forschungsarbeiten zu untersuchen. Die Notwendigkeit dieser Forschungsarbeiten wird jedoch in diesem Beitrag am Beispiel eines falsch verdrahteten Roboters sowie den dadurch verursachten technischen Schäden und finanziellen Verlusten verdeutlicht. Erste Vorschläge wie solche TD zukünftig visualisiert werden können, wurden vorgestellt. Eine eigens durchgeführte Umfrage verdeutlicht, dass der langfristige zusätzliche Aufwand aufgrund einer suboptimalen Lösung den kurzfristigen Nutzen übersteigt. Das daraus gewonnene Ergebnis kann dazu beitragen, dass Industrieunternehmen während der

Inbetriebnahme eines Produktionssystems die Anzahl von TD reduzieren und in bestehenden ähnlichen Systemen entfernen. Somit könnte der Wartungsaufwand reduziert und die Wirtschaftlichkeit der Unternehmen erhöht werden. Um die Klassifikation von TD zu verbessern, konnten wir in Vorarbeiten [4, 5] zwei zusätzliche Fallbeispiele näher untersuchen und planen diese durch weitere Fallbeispiele in Zukunft zu erweitern.

Link zur Umfrage: https://umfrage.ais.mw.tum.de/index.php/947666?lang=de

Literatur- und Quellenverzeichnis

[1] W. Cunningham, "The WyCash portfolio management system," *ACM SIGPLAN OOPS Messenger*, vol. 4, no. 2, pp. 29–30, 1993.

[2] B. Vogel-Heuser and S. Rösch, "Applicability of Technical Debt as a Concept to Understand Obstacles for Evolution of Automated Production Systems," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, 2015, pp. 127–132.

[3] T. Besker, A. Martini, M. Tichy, and J. Bosch, "An investigation of Technical Debt in Automatic Production Systems," in *Proceedings of the XP2017 Scientific Workshops on - XP '17*, 2017. *In press.*

[4] L. Capitán and B. Vogel-Heuser, "Metrics for Software Quality in automated Production Systems as an Indicator for Technical Debt," in *2017 IEEE International Conference on Automation Science and Engineering (CASE)*, 2017. *In press*.

[5] B. Vogel-Heuser and E.-M. Neumann, "Adapting the concept of technical debt to software of automated Production Systems focusing on fault handling, modes of operation, and safety aspects," in *Proceedings of the 20th World Congress of the International Federation of Automatic Control (IFAC)*, 2017. *In press.*

[6] Z. Li, P. Avgeriou, and P. Liang, "A systematic mapping study on technical debt and its management," *Journal of Systems and Software*, vol. 101, pp. 193–220, 2015.

[7] B. Vogel-Heuser, J. Fischer, S. Feldmann, S. Ulewicz, and S. Rösch, "Modularity and architecture of PLC-based software for automated production Systems: An analysis in industrial companies," *Journal of Systems and Software*, vol. 131, pp. 35–62, 2017.

Autoren

Thomas Aicher ist wissenschaftlicher Mitarbeiter am Lehrstuhl für Automatisierung und Informationssysteme der TU München. Sein Forschungsinteresse gilt der modellbasierten Entwicklung und Flexibilisierung von Steuerungssoftware von mechatronischen Systemen.

Quang Huan Dong ist wissenschaftlicher Mitarbeiter am Lehrstuhl für Automatisierung und Informationssysteme der TU München. Sein Forschungsinteresse gilt der Analyse von Technical Debt in mechatronischen Systemen sowie deren kurz- und langfristigen Auswirkungen.

Prof. Dr.-Ing. Birgit Vogel-Heuser leitet den Lehrstuhl für Automatisierung und Informationssysteme der TU München. Sie forscht an der Entwicklung und Systemevolution verteilter intelligenter eingebetteter Systeme in mechatronischen Produkten und Produktionsanlagen. Sie ist Sprecherin des Sonderforschungsbereiches (SFB) 768 „Zyklenmanagement von Innovationsprozessen“ und Mitglied der acatech.

Kontakt

Internet: www.ais.mw.tum.de
Email : {thomas.aicher; huan.dong; vogel-heuser}@tum.de