

Anforderungen an das Software-Engineering in der Automatisierungstechnik

B. Vogel-Heuser¹, C. Diedrich², A. Fay³, P. Göhner⁴

¹Lehrstuhl für Automatisierung
und Informationssysteme,
Technische Universität München
vogel-heuser@ais.mw.tum.de

²Institut für Automation und
Kommunikation e.V., Magdeburg
christian.diedrich@ifak.eu

³Institut für Automatisierungstechnik,
Helmut-Schmidt-Universität Hamburg
alexander.fay@hsu-hh.de

⁴Institut für Automatisierungs- und
Softwaretechnik,
Universität Stuttgart
peter.goehner@ias.uni-stuttgart.de

Abstract: Automatisierte Systeme sind komplexe Hardware-Software-Systeme, die zu einem hohen Maß von der Qualität des zugrundeliegenden Software-Engineering abhängen. Der Beitrag soll durch die Klärung der spezifischen Anforderungen zu einem besseren Verständnis von Automatisierungstechnik und Software-Engineering führen und die Grundlagen für eine engere Zusammenarbeit legen. Es werden die wesentlichen Anforderungen identifiziert, die sich aus dem funktionalen Aufbau sowie aus den Randbedingungen des Lebenszyklus automatisierter Systeme ergeben. Auf dieser Grundlage können Methoden und Werkzeuge verbessert und so die Qualität des Software-Engineering gesteigert werden.

1. Software-Engineering in der Automatisierungstechnik

Die Automatisierungstechnik befasst sich mit der Automatisierung von Systemen, die aus Hardware und einem wachsenden Anteil an Software bestehen. Ein automatisiertes System besteht aus einem technischen Prozess, der in einem technischen System abläuft, das alle zur Automatisierung des technischen Prozesses notwendigen technischen Einrichtungen enthält, dem Automatisierungssystem und dem Prozess- und Bedienpersonal [Gö12], siehe Abbildung 1. Ein technisches System ist dabei entweder ein technisches Produkt oder eine technische Anlage. Technische Produkte sind Massenprodukte mit wenigen Sensoren und Aktuatoren und einem hohen Automatisierungsgrad, wie z.B. Haushaltsgeräte. Verglichen mit technischen Produkten sind technische Anlagen Einmalsysteme mit einer großen Anzahl an Sensoren und Aktuatoren und weisen einen mittleren bis hohen Automatisierungsgrad auf, wie z.B. industrielle Produktionsanlagen.

Aus dem speziellen Aufbau der Systeme und den Randbedingungen bezüglich Prozess, Hardware und Interaktion mit dem Prozess- und Bedienpersonal ergeben sich Anforderungen, die im weiteren Beitrag als zusammengehörige *funktional-vertikale Kategorie* behandelt werden.

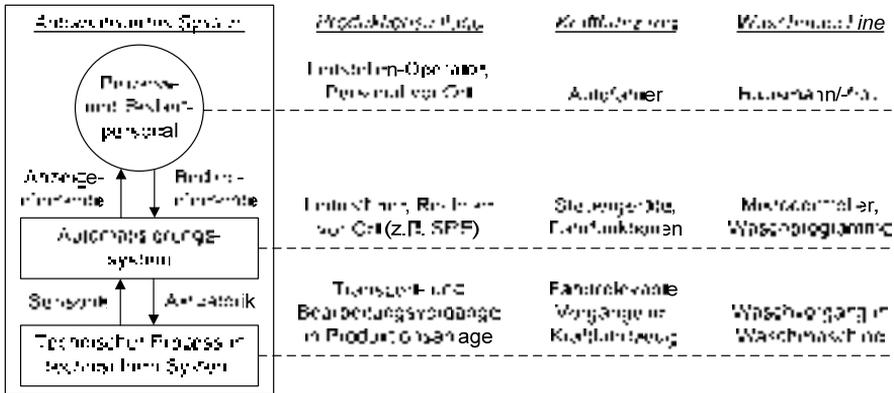


Abbildung 1: Prinzipieller Aufbau eines automatisierten Systems [Gö12] mit Beispielen

Automatisierte Systeme sind heutzutage im Alltag weit verbreitet. Sie unterstützen ihre Nutzer bei langwierigen, schwierigen oder auch gefährlichen Aufgaben und sind aus dem privaten Bereich, dem Arbeitsleben oder in sich überschneidenden Bereichen, wie dem Individual-, Nah- und Fernverkehr nicht mehr wegzudenken. Aufgrund der Allgegenwärtigkeit im direkten Umfeld der Menschen und der gesellschaftlichen Abhängigkeit von diesen Systemen muss gewährleistet sein, dass die Systeme und insbesondere ihre Software und damit auch das Software-Engineering hohen Qualitätsstandards gerecht werden, um Ausfälle, Fehlfunktionen oder Sach- und Personenschäden möglichst auszuschließen. Allerdings lassen die Kombination von Software und Hardware und die Verbreitung der Systeme in den unterschiedlichsten Anwendungsdomänen die Komplexität der Systeme ansteigen und führen beim Software-Engineering zu komplexen Zusammenhängen, die sich immer schwerer beherrschen lassen [WaRa08]. Zudem erfordert die Entwicklung der Systeme die Zusammenarbeit unterschiedlicher Disziplinen, deren zielgerichtete Koordination entscheidendes Erfolgskriterium ist [Ma10]. Umso wichtiger ist in diesem Zusammenhang ein gemeinsames Verständnis der Anforderungen, die beim Engineering der Systeme zu erfüllen sind.

Beim Software-Engineering in der Automatisierungstechnik ergeben sich entscheidende Anforderungen aus dem gesamten Lebenszyklus der automatisierten Systeme - von Beginn des Engineering, während der Laufzeit, bis zum Ende ihrer operativen Betriebsphase. Diese werden im Weiteren ebenfalls als zusammengehörige *zeitlich-horizontale Kategorie* diskutiert.

In diesem Beitrag werden, ausgehend von den beiden genannten Kategorien, die aus heutiger Sicht bestehenden Anforderungen an das Software-Engineering in der Automatisierungstechnik identifiziert. Diese Anforderungen dienen einem besseren Verständnis und damit einer erfolgreichen und konstruktiven Zusammenarbeit von Automatisierungstechnik und Software-Engineering. Der Beitrag gliedert sich in ein Kapitel zu den grundlegenden Zusammenhängen, deren Betrachtung zu einer Verfeinerung der Kategorien in Subkategorien führt. Im dritten Kapitel werden die konkreten Anforderungen in den Subkategorien detailliert erläutert. Abschließend werden die Anforderungen zusammengefasst und die weiteren Herausforderungen diskutiert.

2. Grundlegende Zusammenhänge in der Automatisierungstechnik

Automatisierung bezeichnet die Überführung von Funktionen eines Systems oder einer technischen Anlage von einem manuellen hin zu einem automatischen Ablauf, um diesen zielgerichtet zu beeinflussen. Der Begriff „automatisch“ bezieht sich hierbei auf den Prozess oder die Einrichtung, „der oder die unter festgelegten Bedingungen ohne menschliches Eingreifen abläuft oder arbeitet“, also selbsttätig erfolgt (vgl. [DIN60050-351], Abschnitt 351-21-40). Nach [DIN60050-351], Abschnitt 351-21-43 ist ein technischer Prozess „ein Prozess, dessen physikalische Größen mit technischen Mitteln erfasst und beeinflusst werden“. Automatisierung erfordert daher im Allgemeinen einen Zugriff auf Informationen aus diesem technischen Prozess (über Sensoren) und/oder eine Möglichkeit des Eingriffs in den technischen Prozess (über Aktuatoren).

Mit Anlagenautomatisierung werden die automatisierungstechnischen Funktionen bezeichnet, wie z.B. auch Sensoren und Aktuatoren, die integraler Bestandteil der Anlagenautomatisierung sind. Der Automatisierungsgrad eines Systems oder einer technischen Anlage entspricht der Norm folgend dem „Anteil der selbsttätigen Funktionen an der Gesamtheit der Funktionen“ des Systems oder der technischen Anlage (vgl. [DIN60050-351], Abschnitt 351-21-41). Bei einem Automatisierungsgrad kleiner eins wird ein System oder eine technische Anlage als teilautomatisiert, andernfalls als vollautomatisiert bezeichnet (vgl. [SS86], S. 375), siehe Abbildung 2.

Ein wichtiges Element bei der Automatisierung ist der Nutzer beziehungsweise die Nutzergruppen, die das System entwickeln, einsetzen, warten und pflegen (Abbildungen 1 und 2). Die *Benutzbarkeit* bildet die aus funktionaler Sicht hierarchisch oberste Subkategorie, in der sich Anforderungen in der funktional-vertikalen Kategorie ergeben.

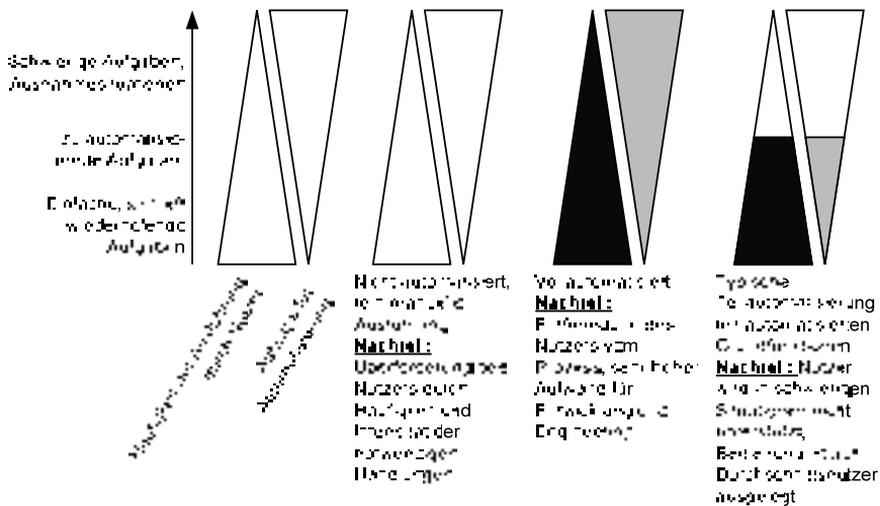


Abbildung 2: Manuelle, voll automatisierte und teilautomatisierte Systeme

Die unterschiedlichen Nutzergruppen beeinflussen den technischen Prozess über die Automatisierungsfunktionen. In Anlagen laufen räumlich verteilt verschiedene Teilprozesse ab (z.B. sequentielle Produktionsschritte). Die dazugehörigen Automatisierungsfunktionen sind daher konzeptionell verteilt und zunächst voneinander unabhängig. Zur Erreichung übergeordneter betrieblicher Ziele (z.B. maximaler Durchsatz) werden übergeordnete Funktionen vorgesehen, welche mit den Automatisierungsfunktionen der einzelnen Produktionsschritte interagieren, was zu weiteren Abhängigkeiten führt.

Die Automatisierungsfunktionen werden heutzutage größtenteils in Software realisiert. Gründe dafür sind die niedrigen Kosten dieser Realisierung und die geringeren Platzanforderungen im Vergleich zu mechanischen oder elektrischen Realisierungen. Vorteilhaft ist bei der Realisierung in Software insbesondere auch, dass die Änderungen und damit Eingriffe während des Betriebs der Anlage kürzere Zeit benötigen als mechanische oder elektrische Umbauten an der Anlage. Außerdem können bei der Signalverarbeitung sehr hohe Genauigkeiten bei ausgesprochener Robustheit gegenüber Umweltbedingungen erzielt werden. Dies ermöglicht auch eine hohe Anpassungsfähigkeit der Funktionen an die spezifischen Bedingungen der einzelnen technischen Prozessstellen.

Die Automatisierungsfunktionen sind das zentrale Element eines automatisierten Systems und haben wesentlichen Einfluss auf die Anforderungen an das Software-Engineering. Die Anforderungen, die sich an die *Automatisierungsfunktionen* ergeben, bilden die zweite Subkategorie der funktional-vertikalen Kategorie.

Die Automatisierungsfunktionen, die in Software realisiert sind, laufen auf speziellen Plattformen ab. Plattform wird hier als Oberbegriff für Hardware als eingebettetes System mit optional Sensor-, Aktuator-, Eingabe- und Anzeigeelementen sowie der Firmware, der kommunikativen Kopplung und hardware-spezifischen Automatisierungsfunktionen verstanden. Abbildung 3 zeigt ausschnittsweise typische Plattformkomponenten, wie sie z.B. bei Antrieben, Bedienpaneelen oder Steuerungen zu finden sind.

Die Berücksichtigung der Eigenschaften der speziellen, in der Automatisierungstechnik eingesetzten Plattformen und Plattformstrukturen ist entscheidend für das Software-Engineering, da nur dann optimal angepasste und qualitativ hochwertige Automatisierungsfunktionen realisiert werden können. Die Anforderungen, die sich durch die *Plattformen* ergeben, bilden die dritte Subkategorie der funktional-vertikalen Kategorie.

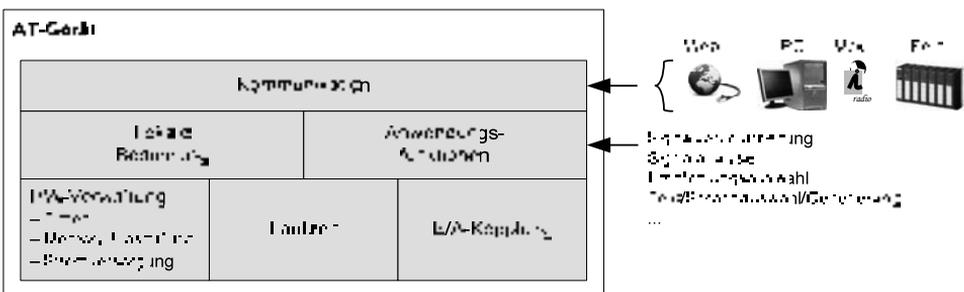


Abbildung 3: Typische Plattformkomponenten eines AT-Geräts.

Betrachtet man schließlich noch die funktional-vertikale Hierarchie als Ganzes, erkennt man, dass sich ein geschlossener Kreis von Abhängigkeiten zwischen Informationserfassung, der Informationsverarbeitung durch die Automatisierungsfunktionen, der Steuer- und Regelsignale der Aktuatoren und der Wirkung im technischen Prozess ergibt. Die Berücksichtigung dieses geschlossenen Kreises, das heißt möglicher Rückkopplungen und Schwingungspotentiale des Systems beim Software-Engineering, ist enorm wichtig, um ein kontrollierbares Verhalten des Systems zu erreichen. Die Anforderungen, die sich durch diesen *geschlossenen Kreis* ergeben, bilden die vierte und letzte Subkategorie in der funktional-vertikalen Kategorie.

Die Definition und Spezifikation neuer Produkte involviert Informationen aus verschiedenen Engineering-Phasen [NA03], z.B. Entwicklung/Konstruktion sowie Betrieb und Wartung ebenso wie aus verschiedenen Engineering-Disziplinen, wie Maschinenbau, Elektrotechnik und Softwaretechnik. Änderungen in einer Disziplin haben Einfluss auf andere Disziplinen. Diese Herausforderungen werden durch die unterschiedlichen Lebensdauern bzw. Änderungszyklen der unterschiedlichen Disziplinen, wie der Mechanik (z.B. Maschine oder Gehäuse), der Elektrotechnik (z.B. automatisierungstechnisches Gerät oder CPU) und der Informationstechnik inklusive der Softwaretechnik erschwert. Während in der Anlagenautomatisierung Maschinen und Anlagen häufig 10 bis 30 Jahre betrieben werden, wird die automatisierungstechnische Hardware im Anlagenbau maximal alle 3 bis 5 Jahre ausgetauscht und die Software maximal alle 6 bis 12 Monate (siehe Abbildung 4, Zeiträume für Produktautomatisierung deutlich kürzer).

Diese Herausforderungen und die zeitlichen Abhängigkeiten im Life-Cycle der Systeme müssen bereits im Software-Engineering berücksichtigt werden, um zur Laufzeit entsprechend flexibel zu sein. Die Anforderungen, die sich an das *Life-Cycle-Management* ergeben, bilden die erste Subkategorie in der zeitlich-horizontalen Kategorie.

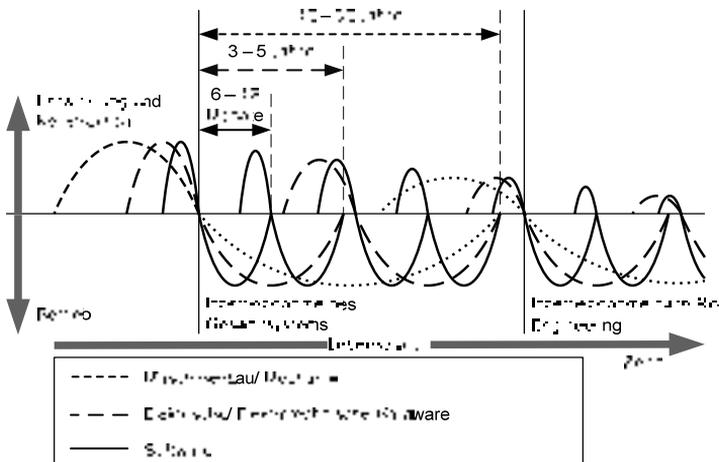


Abbildung 4: Zusammenwirken der verschiedenen Disziplinen in der Anlagen- und Produktautomatisierung in Anlehnung an Li et al. [Li12] (Zeiten für Produktautomatisierung abweichend)

Die besondere Kompetenz der Automatisierungstechniker liegt darin, Methoden aus anderen wissenschaftlichen Disziplinen für die Entwicklung realer technischer Systeme zu einem integrativen Gesamtprozess zu verbinden. Dabei sind im Engineering die besonderen Herausforderungen und Randbedingungen dieser realen technischen Prozesse und Automatisierungssysteme zu berücksichtigen und geeignete und angepasste Methoden erforderlich, um qualitativ hochwertige Ergebnisse zu erreichen.

Dabei basiert die Automatisierungstechnik ganz wesentlich auf Modellen. Modelle werden u.a. für den Entwurf, die Implementierung, den Test, die Optimierung und die Diagnose von Automatisierungssystemen benötigt. In der Automatisierungstechnik werden in der Praxis neben systemtheoretischen Modellen zahlreiche Modelle aus den Fachgebieten verwendet, welche die zu automatisierenden Systeme gestalten. Die automatisierungstechnische Forschung bezieht darüber hinaus Modelle verschiedenster Wissenschaftsdisziplinen, z.B. auch der Informatik, ein und adaptiert diese für die Belange der Automatisierungstechnik. Dabei steht die Automatisierungstechnik immer im Spannungsfeld zwischen der gewünschten Verallgemeinerung der dynamischen Phänomene des zu automatisierenden Systems und der Notwendigkeit, die physikalischen, technischen und auch organisatorischen Gegebenheiten des jeweiligen Anwendungsgebiets zu berücksichtigen, um Modelle, Methoden und schließlich auch Werkzeuge zu entwickeln.

Für den Entwurf und die Realisierung (d.h. Programmierung) von Automatisierungsfunktionen wurden über Jahrzehnte verschiedene Modelle und Beschreibungsmittel entwickelt oder von anderen Disziplinen übernommen und adaptiert. Aus der Mathematik wurden beispielsweise die Boolesche Algebra und die Automatentheorie übernommen, aus der Elektrotechnik der Stromlaufplan (KOP [IEC61131-3]) und die signalorientierte Darstellung der Funktionspläne (FBD [IEC61131-3]). Eine Übersicht über Anforderungen an Beschreibungsmittel zur Steuerungsprogrammierung und eine Bewertung einiger Beschreibungsmittel wurde von der Gesellschaft für Automatisierungstechnik erarbeitet [VDI3681] (siehe Übersicht erweitert um UML-Derivate, Tabelle A1).

Die erforderlichen Eigenschaften der *Modelle* einerseits und der *Werkzeuge* andererseits sind daher entscheidend für das Software-Engineering und bilden die zweite und dritte Subkategorie in der zeitlich-horizontalen Kategorie von Anforderungen.

Die Bemühungen zur Einführung der Automatisierung sowohl in Produktionsprozessen als auch in Prozessen anderer Anwendungsbereiche haben sich in den letzten Jahren grundlegend verändert. Während zunächst die Automatisierung von starren, wiederkehrenden Abläufen im Vordergrund stand, gehen heutige Bestrebungen dahin, flexible Systeme mit unterschiedlichen Aufgabenstellungen einzusetzen. Neben der Installation automatisierter Systeme ist insbesondere auch die Automatisierung von bereits existierenden, jedoch noch nicht automatisierten oder nur teilautomatisierten Prozessen und Abläufen eine wesentliche Aufgabe der Automatisierung. Beispielsweise müssen im Falle von Fehlern in der Anlage Eingriffe in der Software durch Elektriker des Betreibers durchgeführt werden, wie z.B. das Setzen von Ausgängen und die Überbrückung von Sensorabfragen bis zur Behebung eines Hardwarefehlers durch Austausch eines Gerätes (Abbildung 5), um in Betrieb und Wartung den Zeitraum des Anlagenstillstands gering zu halten.

SPS/PLS-Engineering-umgebung aus den Phasen des Engineering	Anforderungsermittlung (Grundlagenermittlung)	Nutzeranforderungen			Systemanforderungen	
	Basisplanung					
	Feinplanung/ Konstruktion					
	Subsystemtest im Werk					
	Inbetriebsetzung	Methoden/ Programmteile von Instanzen bilden				
	Abnahme	Sub-System-Test			Systemtest	
	Monitoring der aktuellen Variablen					
	Betrieb und Wartung (Multiuser)	Programm-änderung online	Forcen von Variablen	„Brücken“	Sicherer Austausch von Programmteilen	Änderung von Variablenzuordnung im Programm
		Re-Engineering: Erweiterung/ Optimierung	Auslesen und Analysieren von bestehenden Programmen		Einspielen geänderter Programmteile	

Abbildung 5: Tätigkeiten im Life-Cycle der Anlagenautomatisierung [Vo09]

Aus diesem Grund ist die Berücksichtigung der Änderbarkeit der Systeme ein wichtiger Faktor im Software-Engineering. Die Anforderungen, die sich an die *Änderbarkeit* ergeben bilden die vierte Subkategorie der zeitlich-horizontalen Kategorie von Anforderungen an das Software-Engineering in der Automatisierungstechnik.

3. Anforderungen an das Software-Engineering

Aus den Bedingungen des Einsatzes von Software in der Anlagenautomatisierung und den grundlegenden Zusammenhängen ergeben sich Anforderungen an die Software, an die sie ausführende Hardware und daraus resultierend, an das Software-Engineering, die in diesem Kapitel, gemäß der Identifizierten Subkategorien gegliedert, erläutert werden.

3.1 Anforderungen an die Benutzbarkeit in der Automatisierungstechnik

Bei der Entwicklung, Nutzung, Wartung und Pflege der Systeme sind viele unterschiedliche Benutzergruppen involviert. Dies führt zu der Anforderung, dass das Software-Engineering all diese unterschiedlichen Personen berücksichtigt und geeignete integrierte Methoden und Prozesse bereitstellt, die ein konsistentes Zusammenarbeiten ermöglichen. Im Einzelnen müssen die folgenden Gegebenheiten berücksichtigt werden:

Im Software-Engineering der Automatisierungstechnik existieren mehrere Ebenen der Software-Erstellung: Zum einen müssen während des Engineering der Automatisierungstechnik die softwaretechnischen Module von Ingenieuren und Informatikern erstellt werden, zum anderen müssen während der Betriebs- und Wartungsphase kurzfristige Änderungen von Technikern und Facharbeitern vorgenommen werden [Vo09].

In den letzten Jahren lässt sich ein Trend im Detail-Engineering erkennen: die Aufteilung in Modulersteller und Modulanwender. Zur Entwicklung von neuen Softwaremodulen für die Modulbibliothek werden häufiger Informatiker eingesetzt, die in der Regel Hochsprachen zur Programmierung bevorzugen und modellbasiert vorgehen wollen.

Für die Erstellung der Applikation werden eher Maschinenbau- oder Elektroingenieure eingesetzt. Diese verwenden vorgefertigte Module und kombinieren diese unter Anwendung der Sprachen der [IEC61131-3]: für die Regelungstechnik Continuous Function

Chart (CFC) und für die Steuerungstechnik in Europa Funktionsbausteinsprache (FBD) bzw. für den Export des Anlagenprojekts ins Ausland im Wesentlichen die Assembler-artige Anweisungsliste (AWL) und der Stromlaufplan-artige Kontaktplan (KOP). Bei anspruchsvollen regelungstechnischen Funktionen bietet sich die Modellierung in MATLAB/Simulink an, die anschließend in IEC 61131-3 Code übersetzt werden muss.

In der Anlagenautomatisierung stehen in der Phase der Wartung und Pflege in Europa in der Regel Facharbeiter oder Techniker zur Verfügung, die auf jeden Fall drei der Programmiersprachen für Speicherprogrammierbare Steuerungen (SPS) beherrschen, die AWL, den KOP und die FBD, während weltweit in der Regel der KOP bevorzugt wird.

3.2 Anforderungen an Automatisierungsfunktionen

Es gibt einen Basiskanon von Automatisierungsfunktionen. Dies betrifft Messen, Stellen und Regeln bzw. Steuern. Bei der generellen Aufgabe der Prozessführung durch die Automatisierung kommen Funktionen wie z.B. Anzeigen, Archivieren, Alarmieren, Beeinflussen und Sichern hinzu. Bei der Umsetzung dieser Funktionen sind Randbedingungen zu beachten, die beispielhaft in Tabelle A2 im Anhang enthalten sind.

Im Vergleich zu Software-Systemen im Bereich Automobil stellen in der Anlagenautomatisierung die Betriebsarten [DIN13128] eine weitere orthogonale Anforderung an die Software-Entwicklung dar. Während diese bei eingebetteten Systemen keine Bedeutung haben, weil diese im Fehlerfall stillgesetzt werden können, muss in der Anlagenautomatisierung das Betreiberpersonal die Anlage im Handbetrieb wieder in einen Zustand fahren, von dem aus diese in den Automatik-Betrieb übergehen und weiter produzieren kann. Dies bedeutet für die Software, dass viele Softwarebausteine nicht nur im Automatik-, sondern auch im Handbetrieb ggf. mit anderen Meldungen, anderen Bedienelementen und reduzierten Geschwindigkeiten betrieben werden müssen und dies in die Software-Entwicklung einbezogen werden muss [FFV12]. Eine Herausforderung der Zukunft ist zudem die geeignete Unterstützung des Operators im Fehlerfall.

Beim Entwurf von Automatisierungs-Software sind zudem weitere Systemaspekte zu berücksichtigen. Die geforderten Automatisierungsfunktionen müssen (im Sinne funktionaler Anforderungen) realisiert werden (dazu gehören bei Produktionsprozessen auch die geforderten Eigenschaften des herzustellenden Produkts). Außerdem müssen die kausalen und zeitlichen Verkopplungen, die über den durch die Automatisierung beeinflussten technischen Prozess und die räumliche Struktur des technischen Systems bestehen (durch Materie- und Energieflüsse sowie Informationen, die diese Flüsse beeinflussen), berücksichtigt werden, um nicht unerwünschte Seiteneffekte oder gar Instabilitäten im System zu erzeugen. Des Weiteren müssen die Eigenschaften derjenigen Komponenten des technischen Systems, die für die Ausführung von Automatisierungsfunktionen unmittelbar relevant sind (Sensoren, Aktuatoren, informationsverarbeitende Komponenten, Kommunikationseinrichtungen) beachtet werden, zum Beispiel hinsichtlich Signallaufzeit, Bearbeitungsgeschwindigkeit, Rechen- und Speicherkapazitäten, Sicherheit und Zuverlässigkeit. Außerdem müssen die automatisierungstechnischen Funktionen eine hohe Anpassungsfähigkeit durch Konfiguration und Parametrierung ermöglichen, um eine wirtschaftliche Stückzahl der Automatisierungskomponenten zu erzielen.

3.3 Anforderungen durch spezifische Plattformen

Die erfolgreiche Umsetzung der fließenden Automatisierung hängt im Wesentlichen davon ab, wie vorhandene, nachgerüstete und neue Anlagen, Maschinen und Komponenten in ein konsolidiertes Automatisierungskonzept überführt werden können. Hierzu muss eine Softwarelösung bereitgestellt werden, die diese Systeme integriert und die Interoperabilität [DLH11] der vorliegenden heterogenen Soft- und Hardwarelandschaft sicherstellt. Im Einzelnen sind die folgenden Gegebenheiten zu berücksichtigen:

Es gibt eine große Vielfalt an Mikroprozessoren und Echtzeitbetriebssystemen. Dabei gibt es keine Vereinheitlichungstendenzen auf diesem Gebiet in der Automatisierungstechnik. Hauptgrund sind die unterschiedlichen funktionalen Umfänge in den Geräten; von einfachen binären Sensoren zu hoch komplexen Antriebsumrichtern und Analysegeräten. Dabei muss die zeitliche Funktionsausführung nach dem Abtasttheorem beachtet werden, da AT-Funktionen oft auf zeitkontinuierliche Systeme wirken. Es gibt parallele synchrone und asynchrone Funktionsausführung in einem Gerät und es müssen Änderungen der Anwendungsfunktionen während des operativen Betriebs möglich sein.

In der Regel erbringen mehrere durch industrielle Kommunikationssysteme gekoppelte Automatisierungsgeräte gemeinschaftlich in einem Abtasttakt eine Funktion, z.B. geschlossener Regelkreis. Hierbei müssen deterministische industrielle Kommunikationssysteme genutzt und eine Synchronisierung über Gerätegrenzen im μs - bis ms -Bereich ermöglicht werden.

Der Zugriff auf die Automatisierungsfunktionen im operativen Betrieb erfolgt parallel von verschiedenen Host-Systemen mit unterschiedlichen zeitlichen und funktionalen Randbedingungen (z.B. Steuerung, Parametrierung und Service oder Diagnose). Daher sind unterschiedliche QoS der Kommunikation in einem Gerät, der konkurrierende Zugriff auf Funktionen von verschiedenen Hosts, die Bedienung unterschiedlicher Funktionalitäten durch Bediener mit unterschiedlichen Rechten und unterschiedlicher Qualifikation und die Bewahrung der Konsistenz der Funktionsparametrierung erforderlich.

Bezüglich der NFA sind vor allem ein geringer Energieverbrauch (im mW -Bereich, z.B. 100mW) für Geräte in bestimmten Einsatzbereichen, zum Teil sehr hohe Anforderungen an numerische Berechnungsgenauigkeiten (z.B. Durchflussberechnung mit 64-Bit-Gleitkommazahlen), die Erfüllung der funktionalen Sicherheit, gemeinsame Kommunikations- und Energieversorgungsleitungen für Geräte und sehr lange Einsatzdauern (10 Jahre und mehr) üblich und erforderlich. Über diese lange Einsatzdauer müssen auch die Nachlieferung von Ersatzteilen und Softwareanpassungen jederzeit möglich sein.

Domänenspezifische Sprachen zur Konfiguration, Parametrierung und Programmierung der AT-Geräte sind entscheidend für den Einsatz in den vielfältigen Anwendungsbereichen. Zudem sind die funktionale Modularisierung und die Hierarchisierung für die Gewährleistung von Determinismus in der Abarbeitung trotz komplexer Zusammenhänge in den teils sehr großen Systemen (mit mehr als 10.000 I/O-Signalen) wesentlich. Zudem sind Technologien zur AT-Geräteintegration und damit auch Interoperabilität der kooperativen Funktionen und sehr langen Gerätelebenszyklen von herausragender Bedeutung [Di09].

3.4 Anforderungen durch den geschlossenen Kreis

Aus der Tatsache, dass in der Anlagenautomatisierung ein technischer Prozess (die technische Strecke) geregelt und gesteuert werden muss, ergeben sich weitere Anforderungen. Sensoren liefern dazu Informationen über physikalische Größen, die von der Automatisierungs-Software laufend erfasst und mit dem Ziel verarbeitet werden, diese (und ggf. andere) physikalischen Größen wiederum über Aktuatoren zu manipulieren.

Aus dieser Struktur ergibt sich unmittelbar die besondere Bedeutung einiger NFA für Automatisierungs-Software: Die Echtzeitfähigkeit ist essenziell, da die physikalische Umgebung nicht warten kann wie ein menschlicher Benutzer. Betriebssicherheit ist ein weiteres Kriterium, da die automatisierten technischen Systeme in der Lage sind, erhebliche Schäden anzurichten. Eine für Außenstehende häufig nicht unmittelbar erkennbare Konsequenz aus der oben beschriebenen Struktur ist, dass die Anforderungserfassung für diese Art von Automatisierungs-Software grundsätzliche Unterschiede zur Anforderungserfassung „üblicher“ Software aufweist. In der Automatisierungstechnik steht nicht das Verhalten der zu entwerfenden Software im Mittelpunkt, sondern das Verhalten des automatisierten Systems. Das ist etwas anderes als die Einbeziehung der Hardware, auf der die Software läuft, es um ein Verhalten geht, das sich durch die Dynamik des beeinflussten Systems und dessen Rückkopplung mit der Automatisierungs-Software ergibt.

Die meisten Anforderungen müssen daher zunächst für das Gesamtsystem erfasst werden, erst dann können die Anforderungen an die Automatisierungs-Software abgeleitet werden. Zum Beispiel bestehen die Anforderungen bei einem Geschwindigkeitsregelsystem für Fahrzeuge zunächst aus Gütemaßen für den geschlossenen Regelkreis (z.B. maximaler Overshoot oder eine maximale Anstiegszeit in der Antwort auf eine sprungförmige Änderung der Sollgeschwindigkeit). Diesen steht die Dynamik des Fahrzeuges gegenüber („Wie groß ist die Trägheit aufgrund der Masse und die Beeinflussbarkeit durch Motor und Bremse?“). Erst aus der Gesamtbetrachtung beider Teile, bei der häufig Modelle der Strecke erstellt und analysiert werden müssen, können die Anforderungen an die Software abgeleitet werden (zum Beispiel, dass ein PID-Regelalgorithmus mit entsprechenden Parametern eingesetzt werden soll oder dass bestimmte Antwortzeiten eingehalten werden müssen). Dieses Vorgehen unterscheidet sich wesentlich von sonst typischen Anforderungserfassungen in der Informatik, da hier Wissen erfasst und methodisch verarbeitet werden muss, das sich nicht durch Kundenbefragung ermitteln lässt, sondern aus der Anwendung automatisierungs- und regelungstechnischer Analyseverfahren entsteht.

3.7 Anforderungen an das Life-Cycle-Management

In der Anlagenautomatisierung insbesondere im Anlagenbau, gilt es den Stillstand der Anlage durch den Austausch der Software so gering wie möglich zu halten bzw. ganz zu vermeiden. Daraus resultiert die Anforderung der Automatisierungstechnik, die Änderungen der Software während der Laufzeit also des Betriebs der Anlage durchzuführen und auch Teile der Hardware, z.B. Sensoren und Aktuatoren oder Ein- und Ausgangsbaugruppen, während der Laufzeit tauschen zu können sowie die Kompatibilität von Software über entsprechend lange Zeiträume zu gewährleisten.

Wie bereits eingangs erläutert, wirken eine Vielzahl unterschiedlicher Komponenten – Hardware wie Software – in der Automatisierungstechnik zusammen. Dabei differieren die Lebens- und Innovationszyklen dieser Komponenten erheblich. Die Zeit in der eine Anlage betrieben wird ist hochgradig abhängig vom Industriezweig [Sc11]. In der Anlagenautomatisierung ist die Sicherstellung der einwandfreien Funktion der eingebauten Komponenten aufgrund des deutlich kürzeren Life-Cycle der Einzelkomponenten – dies gilt insbesondere für Software – eine große Herausforderung. Zusätzlich werden die Systeme in Bezug auf ihre Funktionalität und Preisgestaltung stetig weiterentwickelt.

Das in [ZVEI10] für den Einsatz in der Automatisierung entwickelte generische Life-Cycle-Modell unterscheidet zwischen Typen und Instanzen von Produkten. Ein Typ stellt dabei ein Produkt dar, das spezifische Anforderungen erfüllt und das durch Integration neuer Funktionen oder durch Einführung neuer Technologien weiterentwickelt wird. Somit entstehen neue Versionen mit unterschiedlichen Eigenschaften. Jede gefertigte Einheit eines Typs bildet eine Instanz dieses Typs und kann durch eine eindeutige Kennung (z.B. Seriennummer) identifiziert werden. Jede Instanz eines Produktes besitzt eine Lebenszeit (Product Life Time), die vom Zeitpunkt ihrer Erzeugung bis zur Entsorgung reicht. Für Softwareprodukte bedeutet dies, dass verschiedene Versionen parallel genutzt und gepflegt werden müssen. Hierzu ist ein entsprechend leistungsfähiges Versionsmanagement erforderlich, das nach klaren Regeln eindeutige Versionskennungen vergibt. Bei der Definition einer solchen Versionsnomenklatur ist insbesondere der Kompatibilität der Komponenten herausragende Bedeutung beizumessen. Ein Kompatibilitätsmodell auf Basis der [DIN62402] ist in [Sc11, ZVEI10] enthalten. Das Fokussieren aller Beteiligten der Wertschöpfungskette auf das Life-Cycle-Management, die Verstärkung proaktiver Betrachtungen bereits ab der Planungsphase einer Anlage auf der Basis eines gemeinsamen Modells ist eine wesentliche Voraussetzung zur Minimierung der Total Cost of Ownership (TCO).

3.5 Anforderungen an Modelle in der Automatisierungstechnik

Vor diesem Hintergrund wurden in der automatisierungstechnischen Forschung bereits zahlreiche Modell-Beschreibungsmittel entwickelt bzw. aus anderen Wissenschaftsdisziplinen übertragen, adaptiert, weiterentwickelt und kombiniert, so dass eine große Vielfalt entstanden ist.

Die Automatisierungstechnik erfordert folgende Teil-Modelle:

1. Modell des zu automatisierenden Systems, welches seine Struktur und sein Verhalten (Dynamik) beschreibt;
2. Modell der Anforderungen (an Struktur und Dynamik, von funktionalen Anforderungen und nichtfunktionalen Anforderungen);
3. Modelle vorhandener Automatisierungs(teil-)lösungen, die eingesetzt werden sollen;
4. Modelle von Vorgehensweisen zur Konzeption, Ausgestaltung und Realisierung von Automatisierungssystemen;
5. Modelle des automatisierten Gesamtsystems, einschließlich der Überprüfungen, in welcher Weise die Anforderungen erfüllt wurden.

Die Modelle wirken wie in Abbildung 6 gezeigt zusammen: aus den Modellen 1, 2 und 3 entsteht das Modell 5 durch Anwendung der Modelle 4. Diese Modelle können unterschiedlichen Formalisierungsgrad haben: zum Einsatz kommen informelle, semi-formale und formale Modelle. Den Vorzügen formaler Modelle hinsichtlich Analysierbarkeit etc. muss immer der erhebliche Aufwand der Erstellung dieser Modelle für reale Systeme (deren Modelle große Ausmaße annehmen) gegenübergestellt werden.

3.6 Anforderungen an Werkzeuge für das Software-Engineering

Für die Erstellung und Auswertung der Modelle werden geeignete Software-Werkzeuge benötigt. Ein wesentlicher Aspekt ist die Kopplung zu den Werkzeugen der Mechanik, der Elektrotechnik sowie den Werkzeugen in den vorgelagerten Phasen des Lebenszyklus und aufgrund der Notwendigkeit der Änderungen im Betrieb auch der nachgelagerten Phasen. Die im Software-Engineering üblichen Model-Driven Engineering (MDE)-Ansätze mit der Generierung von Code und der ausschließlichen Änderungen im Modell sind in der Anlagenautomatisierung nicht durchsetzbar [VBK11], weil auf den Baustellen weltweit Änderungen im Code auf der Zielplattform schnell durchgeführt werden müssen, eine Rücksprache mit einer im schlimmsten Fall 12 h versetzten Konstruktionsabteilung oder eine Wartezeit aber nicht akzeptabel sind.

Entsprechend der NFA aus dem Lebenszyklus, der Domänenintegration sowie der Nutzer sind folgende Anforderungen an die Werkzeuge hervorzuheben:

Die Werkzeuge müssen über eine lange Zeit verfügbar sein (Reliability → Availability, Maintainability, Portability) und auf die Benutzergruppen zugeschnitten sein (Usability, Functional Suitability). Sie müssen modulare Planungs- und Entwicklungsprozesse unterstützen, da AT-Systeme aus vorgefertigten Systemen zusammengesetzt werden. Deshalb müssen Werkzeuge Bibliothekskonzepte von Komponententypen unterstützen (Maintainability → Reusability) und besonders Konfigurationsprozesse unterstützen. Vor allem für die integrative und erfolgreiche Zusammenarbeit aller Beteiligten ist zudem die Interoperabilität [DFB11] der eingesetzten Software-Werkzeuge und die Unterstützung von Varianten- und Versionsmanagement im Sinne des Life-Cycle Managements unerlässlich.

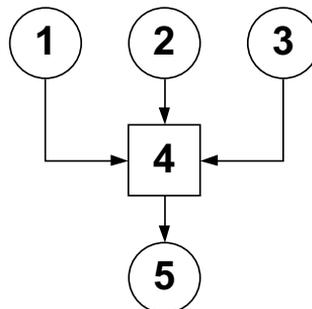


Abbildung 6: Erstellung eines Gesamtsystemmodells für Automatisierungs-Software

3.8 Anforderungen an die Änderbarkeit zur Laufzeit

Änderungen zur Laufzeit sind insbesondere für verfahrenstechnische Prozesse zwingend. Im Bereich des Maschinen- und Anlagenbaus gewinnt der Subsystemtest im Werk immer mehr an Bedeutung, um Inbetriebnahmezeiten auf der Baustelle zu reduzieren. Die Inbetriebsetzung besteht aus Ein-/Ausgangsprüfung sowie der eigentlichen Inbetriebnahme der Subsysteme bis zum Gesamtsystemtest, der durch die Abnahme abgeschlossen wird. Als wesentliche Funktionalität während der Inbetriebsetzung wird das Bilden von Instanzen von Methoden bzw. Programmteilen aus Sicht des Maschinenbaus angestrebt (bei objektorientierter Programmierung), wenn nicht der Applikationsingenieur selbst die Inbetriebnahme durchführt, weil es sich um einen Prototypen einer Anlage oder Teilanlage handelt. Für die Betriebs- bzw. Wartungsphase sind das Monitoring von aktuellen Variablen und die Onlinemanipulation des Programms, wie das Forcen von Variablen oder auch der sichere Austausch von Programmteilen bzw. das sichere Ändern von Variablenzuordnungen im Programm zu unterstützen. Wesentlich ist dabei auch der Multiuser-Modus. Für das Re-Engineering bzw. die Optimierung der Anlage sind insbesondere das Auslesen und das Analysieren von bestehenden Programmen wesentlich. Die Fähigkeiten der Online-Beobachtung und -Änderungen ist eines der Alleinstellungsmerkmale der SPS im Vergleich zu IPCs und der Hochsprachenprogrammierung. Der finale Test der Software erfolgt in der Regel vor Ort auf der Anlage, da nur dort die Randbedingungen vorhanden sind, um alle Aspekte der Funktion zu prüfen, z.B. die Luftfeuchte, die für viele Prozesse relevant ist, oder das Rohmaterial für die Fertigung.

4. Zusammenfassung und Ausblick

Der vorliegende Beitrag gibt eine erste Einführung der wesentlichen Anforderungen an die Software und das Software-Engineering in der Anlagenautomatisierung. Die Anforderungen werden dabei aus zwei Sichten betrachtet. Zum einen werden die funktionalen Eigenschaften, Randbedingungen und Gegebenheiten betrachtet und aus dem vertikalen Aufbau der Systeme, vom Bediener bis zum technischen Prozess, einzelne Subkategorien von Anforderungen gebildet. Innerhalb dieser Subkategorien – den Anforderungen an die Benutzbarkeit, an die Automatisierungsfunktionen, den Anforderungen durch die verwendeten Plattformen und den geschlossenen Kreis – werden die einzelnen Anforderungen detailliert beschrieben. Zum anderen ergeben sich Anforderungen aus Gegebenheiten zu unterschiedlichen Zeitpunkten in den einzelnen Phasen des Life-Cycle der automatisierten Systeme. Diese Anforderungen werden zeitlich-horizontal in Subkategorien – Anforderungen an das Life-Cycle-Management, an die Modelle, an die Werkzeuge und an die Änderbarkeit zur Laufzeit – unterteilt und ebenfalls detailliert erläutert.

Viele der Ansätze aus dem allgemeinen Software-Engineering, wie die Objektorientierung oder MDE-Ansätze, sind bereits in die Anlagenautomatisierung eingeflossen, stoßen aber an ihre Grenzen, wenn sie nicht an die Anforderungen adaptiert werden. Der Beitrag möchte die Diskussion und die Kooperation zwischen Informatik und Automatisierungstechnik auf Basis eines besseren gegenseitigen Verständnisses verstärken.

Literaturverzeichnis

- [DLH11] Diedrich, C.; Lüder, A.; Hundt, L.: Bedeutung der Interoperabilität bei Entwurf und Nutzung automatisierter Produktionssysteme. *Automatisierungstechnik (at)*, Vol. 59, Nr. 7, 2011, S 426–438.
- [Di09] Diedrich, C.: Feldgeräte-Instrumentierungstechnologien – Integration intelligenter Feldgeräte in PLS. In: *Handbuch der Prozessautomatisierung*, München. Oldenbourg Industrieverlag, 2009, S. 290–322.
- [DIN13128] DIN EN 13128: Sicherheit von Werkzeugmaschinen – Fräsmaschinen (einschließlich Bohr-Fräsmaschinen). 2009.
- [DIN60050-351] DIN IEC 60050-351: Internationales Elektrotechnisches Wörterbuch – Teil 351: Leittechnik. 2009.
- [DIN62402] DIN EN 62402: Anleitung zum Obsoleszenzmanagement. 2007.
- [DFB11] Drath, R.; Fay, A.; Barth, A.: Interoperabilität von Engineering-Werkzeugen: Konzepte und Empfehlungen für den Datenaustausch zwischen Engineering-Werkzeugen. *Automatisierungstechnik (at)*, Vol. 59, Nr. 3, 2011, S. 451-460.
- [FFV12] Fuchs, J.; Feldmann, S.; Vogel-Heuser, B.: Modularität im Maschinen- und Anlagenbau – Analyse der Anforderungen und Herausforderungen im industriellen Einsatz. In: *EKA, Magdeburg*, 2012, S. 306–317.
- [Gö12] Göhner, Peter: *Automatisierungstechnik I, Skript zur Vorlesung Automatisierungstechnik I*, Institut für Automatisierungs- und Softwaretechnik, Universität Stuttgart, 2012.
- [IEC61131-3] IEC 61131-3: Speicherprogrammierbare Steuerungen – Teil 3: Programmiersprachen. 2009.
- [Li12] Li, F. et al.: Specification of the Requirements to Support Information Technology-Cycles in the Machine and Plant Manufacturing Industry. In: *14th IFAC INCOM Symposium, Bukarest, Rumänien*, 2012.
- [Ma10] Maga, C. et al.: Mehr Systematik für den Anlagenbau und das industrielle Lösungsgeschäft – Gesteigerte Effizienz durch Domain Engineering. *Automatisierungstechnik (at)*, Vol. 58, Nr. 9, 2010, S. 524-532..
- [NA03] NA 35: Abwicklung von PLT-Projekten. *Arbeitsblatt Namur*. 2003.
- [Sc11] Schrieber, R. et al.: Kompatibilität: der zentrale Schlüssel für Nachhaltigkeit – Life-Cycle-Excellence durch proaktives Handeln. *Automatisierungstechnische Praxis (atp)*, Vol. 53, Nr. 11, 2011, S. 50–57.
- [SS86] Spur, G.; Stöferle, T.: *Handbuch der Fertigungstechnik – Fügen, Handhaben und Montieren*. München, Hanser Verlag, 1986.
- [VBK11] Vogel-Heuser, B.; Braun, S.; Kormann, B.: Implementation and evaluation of UML as modeling notation in object oriented software engineering for machine and plant automation. In: *IFAC World Congress, Mailand*, 2011, S. 9151 – 9157.
- [VDI3681] VDI/ VDE 3681: Einordnung und Bewertung von Beschreibungsmitteln aus der Automatisierungstechnik. 2005.
- [Vo09] Vogel-Heuser, B. (Hrsg.): *Automation & Embedded Systems – Effizienzsteigerung im Engineering*. Kassel University Press, 2009.
- [WaRa08] Wahlster, W.; Raffler, H. (Hrsg.): „Forschen für die Internet-Gesellschaft: Trends, Technologien, Anwendungen“, *Feldafinger Kreis, Feldafing*, 2008.
- [ZVEI10] ZVEI e.V.: *Life-Cycle-Management für Produkte und Systeme der Automation*. Frankfurt, 2010, Online: <http://www.zvei.org/Verband/Publikationen/Seiten/Life-Cycle-Management-für-Produkte-und-Systeme-der-Automation.aspx>.

Tabelle A2: Aspekte von Automatisierungsfunktionen (DE-Domänenexperte)

	Hardware-Voraussetzungen	Zeitliche Voraussetzungen	Beschreibungsmittel	Nutzer Human Machine Interface	...
Messen	an Sensor/Messumformer gebunden, Timer für Laufzeitüberwachung	Abtasttheorem	Eingebettetes System bei Entwicklung	Anlagenfahrer, Betriebspersonal Servicepersonal	
Stellen, Schalten	an Aktuator gebunden, Timer für Laufzeitüberwachung	Abtasttheorem	Eingebettetes System bei Entwicklung	Anlagenfahrer, Betriebspersonal Servicepersonal	
Regeln, Steuern	nicht Hardware gebunden, Timer für Laufzeitüberwachung	Abtasttheorem	programmierbar durch Meister, Techniker, AT- oder DE, z.B. IEC 61131-3	Anlagenfahrer, Betriebspersonal Servicepersonal	
Archivieren	nicht Hardware gebunden	mit Zeitstempel	wie IT Systeme	Anlagenfahrer, Betriebspersonal Servicepersonal	
Registrieren	nicht Hardware gebunden	mit Zeitstempel	wie IT Systeme	Anlagenfahrer, Betriebspersonal Servicepersonal	
Anzeigen	Industrietauglichkeit	in menschlicher Dynamik (1/2 ms)	konfigurierbar durch Meister, Techniker, AT- oder DE	Anlagenfahrer, Betriebspersonal Servicepersonal	
Beeinflussen	Industrietauglichkeit	in menschlicher Dynamik (1/2 ms)	konfigurierbar durch Meister, Techniker, AT- oder DE	Anlagenfahrer, Betriebspersonal Servicepersonal	
Sichern	Besondere Anforderungen	Abtasttheorem	Spezialisten, TÜV, IEC 61508	Anlagenfahrer, Betriebspersonal Servicepersonal	