

Kompetenz in Mechatronik

Software und Systems Engineering

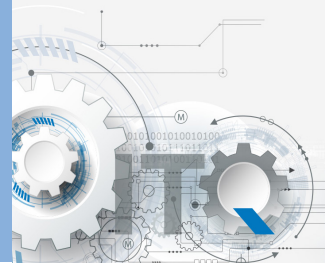


Vision oder Wahrheit?

Software mit Maschine oder Maschine mit Software

„Menschen mit einer neuen Idee gelten so lange als Spinner,
bis sich die Sache durchgesetzt hat“

Mark Twain



Denken in Systemen

Software mit Maschine: Heute mehr denn je – und doch erst am Anfang

Als wir vor mehr als 15 Jahren den Fachverband Software im VDMA mitbegründet haben, um den steigenden Stellenwert gerecht zu werden, musste eigens die Satzung des VDMAs geändert werden, um überhaupt Unternehmen mit „Software-Hintergrund“ als VDMA-Mitglieder aufnehmen zu können.

Inzwischen ist es aber unstrittig, dass der damals als provokant empfundene Leitsatz: in Zukunft wird es nicht mehr **„Maschine mit Software“** sondern **„Software mit Maschine“** geben, vom Grundsatz richtig war. Dies wird insbesondere im Zeitalter von Industrie 4.0 klar, in dem man statt Software mit Maschine „Cyber-Physical System“ sagt.

Dieser technologische Wandel hat die Form des (Systems) Engineerings stark beeinflusst, daher muss die Software nicht – wie in der Vergangenheit –

einfach nur programmiert, sondern systematisch entwickelt werden.

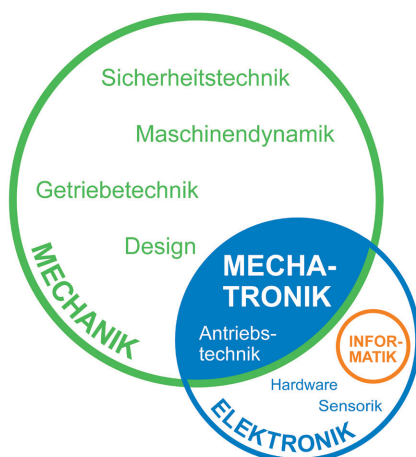
Der Paradigmenwechsel hin zu immer mehr Software-Funktionen wird sich auch in Zukunft nicht verlangsamen. Um diesen Wandel wirklich zu beherrschen, bedarf es einer praxisorientierten und den gegebenen technischen Anforderungen angepassten Berufsausbildung in Unternehmen und Hochschulen/Universitäten.

Die drei Handlungsfelder

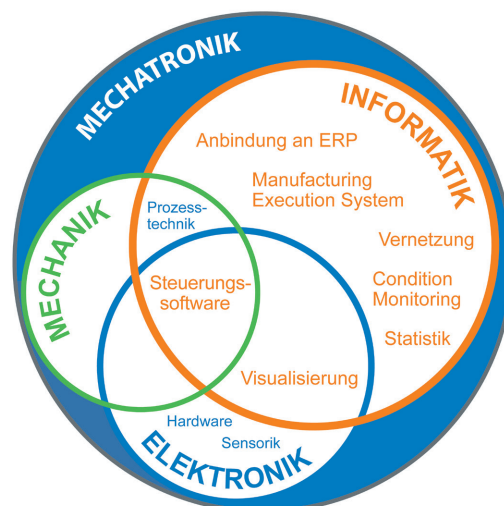
- Mechatronisches Systems Engineering
- agiles/strukturiertes Software Engineering
- mechatronische Ausbildung für Jung bis „Alt“

stellen die zentralen Eckpfeiler von ITQ dar und werden im folgenden detailliert dargestellt.

MECHATRONIK
EINE TEILAUFGABE



MECHATRONIK
DIE GANZHEITLICHE AUFGABE



Unsere Handlungsfelder

Mehr als nur Programmieren

Vielen unserer Kunden fehlen häufig die erforderlichen Entwicklungs-Kapazitäten, um alle Anforderungen des Marktes zu erfüllen. Deshalb unterstützen wir unsere Kunden in vielen Fällen bei der Implementierung der Software.

Wenn wir ein Umsetzungs-Projekt durchführen, stellen wir oft fest, dass die Anforderungen an die Software nicht wirklich klar sind. Aus diesem Grund unterstützen wir nicht nur in der Implementationsphase, sondern auch während der Analysephase, beim Design der Software sowie bei der Durchführung der Software-Qualitätssicherung. Eine klare Spezifikation der Anforderungen und ein modulares Design der Software sind wichtige Voraussetzungen, um eine systematische Qualitätssicherung durchführen zu können.

Wir unterstützen unsere Kunden aber nicht nur beim Software Engineering, sondern begleiten sie auch bei der Einführung des mechatronischen Systems Engineerings. In diesen Projekten beraten wir die Kunden, wie sie die Organisation anpassen müssen und welches die für die jeweilige Situation am besten geeigneten Technologien und Methoden sind.

Das Thema mechatronische Ausbildung ist für uns von zentralem Stellenwert. In Deutschland gibt es einen großen Mangel an Ingenieuren, deshalb beschäftigen wir uns nicht nur damit, wie man Studenten möglichst effizient ausbildet, sondern bieten eine Vielzahl von Aktivitäten, um Jung und Alt für technische Zusammenhänge zu begeistern und weiterzubilden.

Software Engineering

- Analyse
- Design
- Implementierung
- Test und Qualitätssicherung
- Software
 - ▶ zur Steuerung (HMI, PLC, Embedded)
 - ▶ zur Verarbeitung (PLM, MES, Fernwartung)

Systems Engineering

- Mechatronische Organisationsberatung
- Benchmark von Technologien und Tools
- Mechatronische Modularisierung
- Projektmanagement
- Mechatronisches Systems Engineering
 - ▶ Projekte in time, on budget
 - ▶ Verkürzung von Inbetriebnahme

Mechatronische Ausbildung

Management



Ingenieure



Studenten

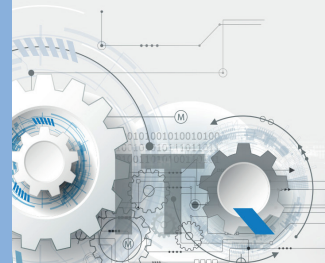


Schüler



Kinder





Vorgehensweise

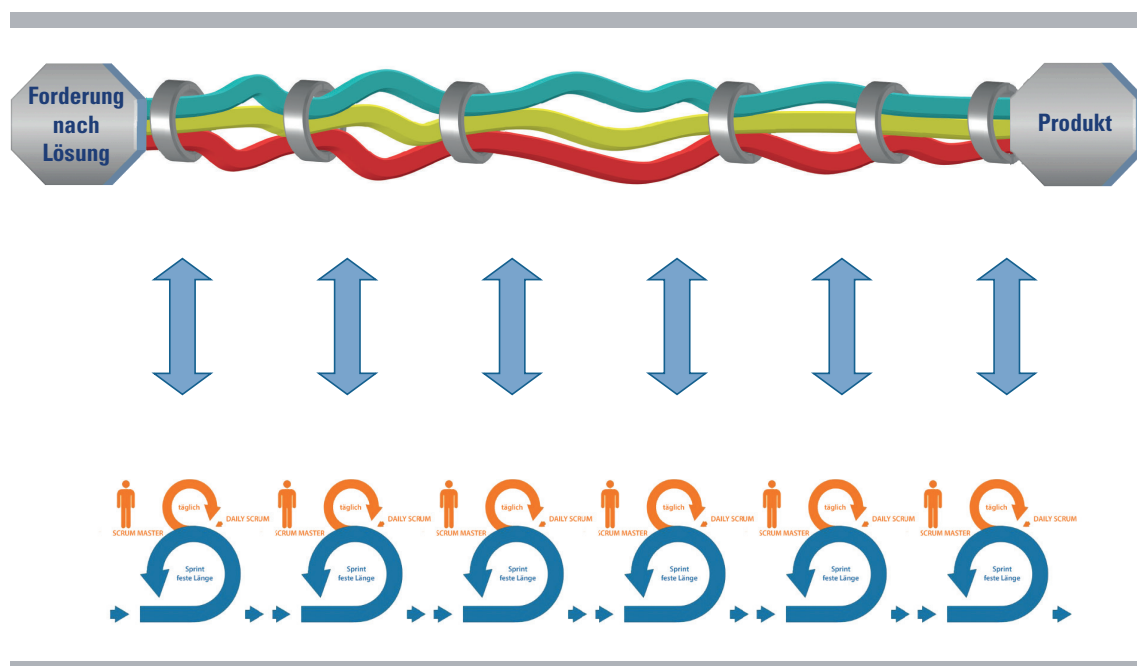
Geordnet, aber trotzdem agil

Kerngedanke der von uns angewandten mechanischen Entwicklungsweise stellt das vom VDMA empfohlene Vorgehen nach dem Quality-Gate-Modell dar, welches die Elemente eines „gesunden disziplinübergreifenden Entwicklungsprozesses“ beschreibt.

Mit dem Ziel der Effizienzsteigerung von Entwicklungsprozessen werden alle Projekthalte auf Grundlage gängiger Techniken des Projektmanagements in ein durchgängiges Vorgehen integriert. Die innerhalb des Entwicklungsprozesses stetig wiederkehrenden Quality Gates verfolgen dabei das Ziel, die erzielten Ergebnisse der beteiligten Disziplinen fortlaufend zu synchronisieren und somit auf Grundlage konkreter Produktanforderungen abzusichern.

In Analogie zu agilen Ansätzen zur Entwicklung von komplexen Softwaresystemen werden Projekte in eine Abfolge von „Sprints“ aufgeteilt. Die Ergebnisse der Sprints werden dabei an jedem Quality Gate in interdisziplinären Workshops einem Review unterzogen und bewertet. Das Ergebnis der Bewertung – dargestellt mittels einer einfachen Ampel-Systematik – gibt dann die Anforderungen und Ziele der nächsten Iteration vor. Zur Durchführung der Reviews werden klare Checklisten definiert.

Mit dieser Kombination aus einem zeitlich stringent vorwärts gerichteten Quality-Gate-Ansatz und einer agilen und iterativen Entwicklungsmethodik erreicht man sowohl Klarheit als auch Flexibilität in komplexen Projekten.



Status Quo

Traditionelles, sequentielles Engineering

Die heutige Form des Engineerings, so wie man sie noch in vielen Unternehmen findet, ist sehr traditionell geprägt. Zu Beginn eines Projektes werden von internen Verantwortlichen oder externen Kunden die Anforderungen in der Regel recht grob und unscharf formuliert.

Danach werden diese Anforderungen meist mit (Vertriebs-)Mitarbeitern des Unternehmens diskutiert, die sehr häufig entweder Konstrukteure sind oder waren. In dieser Abstimmungsphase werden oftmals schon erste Ideen für konstruktive Lösungen diskutiert.

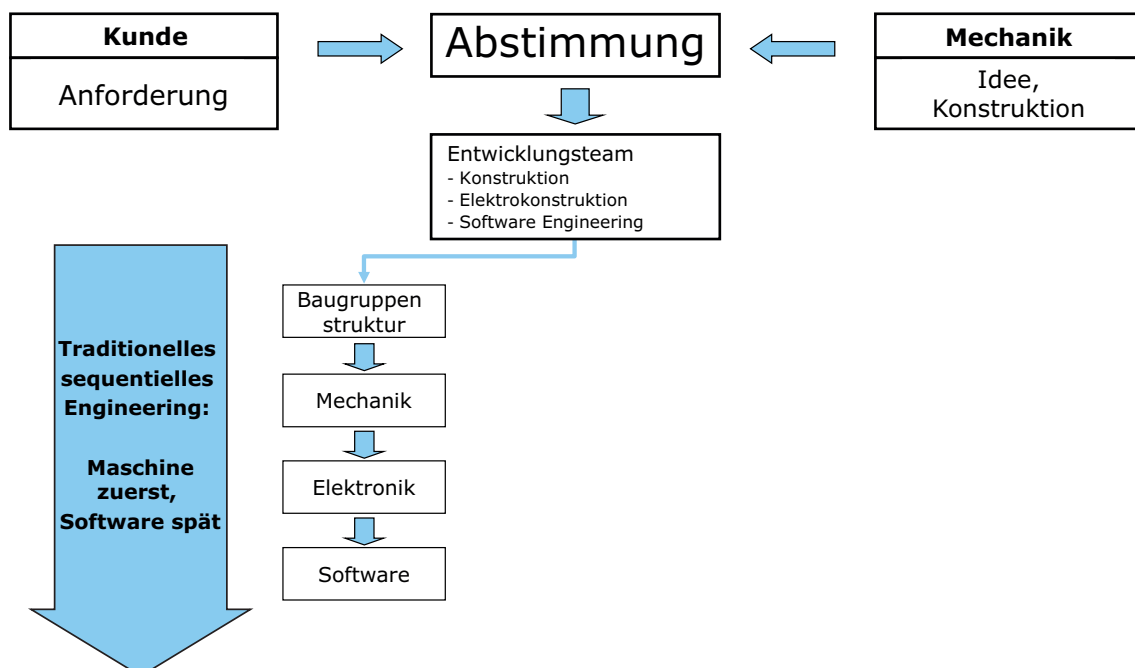
Das Team zur Umsetzung setzt sich dann in der Regel aus mechanischen und elektrischen Konstrukteuren und Software-Ingenieuren zusammen. Basis für deren Arbeiten ist die Liste der vorhandenen

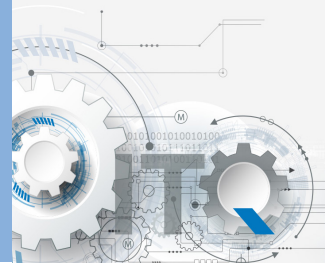
Baugruppen. Dann werden zunächst die mechanischen Konstruktionen durchgeführt und danach wird die Elektrotechnik bestimmt. Meist recht spät im Projektverlauf wird die dazu notwendige Softwareprogrammierung initiiert und mit der Umsetzung der Software begonnen.

Den Ablauf dieser traditionellen Form des Engineerings kann man also folgendermaßen zusammenfassen:

„Die Maschine ist fertig, jetzt fehlt nur noch die Software“.

Wie wir alle wissen, bringt diese sequentielle Form des Engineerings allerdings auch einige Nachteile mit sich, insbesondere in einer Zeit, in der der Softwareanteil in den Maschinen stetig ansteigt.





Das Ziel

Interdisziplinär abgestimmtes Systems Engineering

Deshalb müssen wir gemeinsam darüber nachdenken, wie in Zukunft ein über die Disziplinen abgestimmtes Engineering aussehen soll.

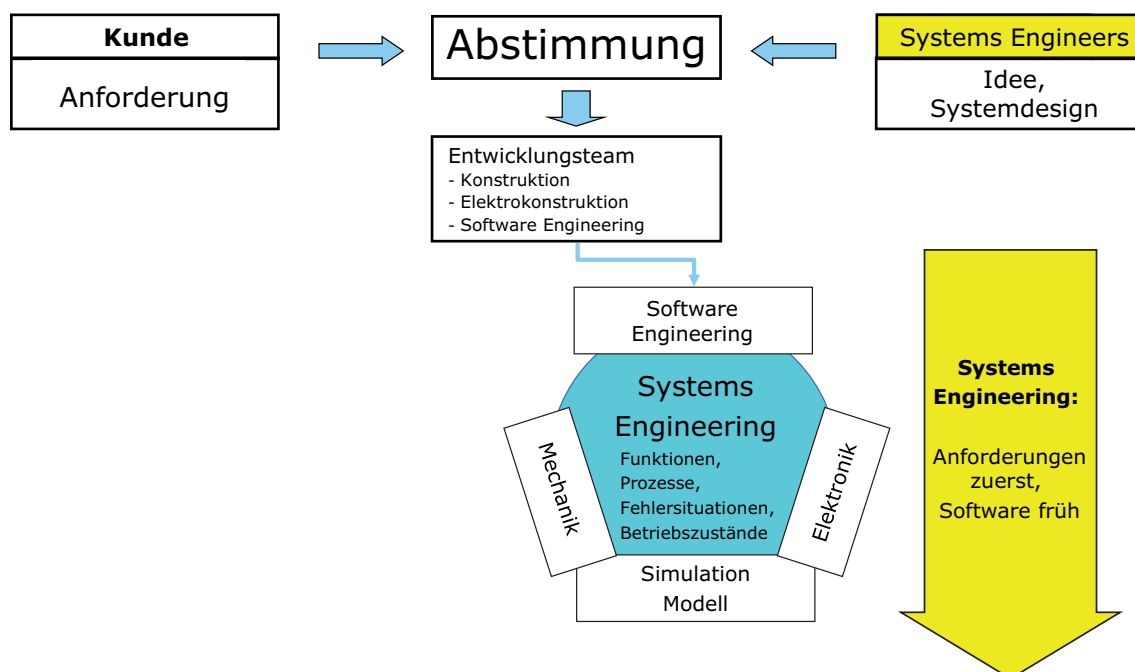
Damit sich ein Prozess signifikant verändert, müssen bereits in der Anfangsphase eines Projektes sogenannte „Systems Engineers“ nicht nur über konstruktive Ideen nachdenken, sondern auch darüber, wie das ganze System inklusive der Software funktioniert.

In den frühen Phasen des Projekts müssen also die Funktionen und Prozesse des Systems definiert werden. Des Weiteren muss festgelegt werden, welche Fehlersituationen in dem System auftreten und wie es darauf reagieren soll. Darüber hinaus muss bestimmt werden, welche Funktionen und Prozesse in welchen Betriebsmodi

zur Verfügung stehen. Wenn diese technischen Anforderungen klar sind, können bereits Arbeiten bzgl. der Software starten. Um die Funktionsfähigkeit der Software zu überprüfen, sollen Simulationsmodelle zum Einsatz kommen. Erst wenn diese Vorarbeiten abgeschlossen sind, sollten die Arbeiten an den mechanischen und elektrischen Komponenten beginnen.

Diese Form des mechatronischen Systems Engineering lässt sich also zusammenfassen in der Formulierung, dass **zuerst die funktionalen Anforderungen (detailliert) zu klären sind, um dann früh mit der Software zu beginnen.**

Diese neue Form des Engineerings fordert nicht nur neue Abläufe in den Unternehmen, sondern auch agile aber strukturierte Vorgehensweisen.



Agile Entwicklungsmethodik

Schneller und flexibler Handeln

Agile Methoden haben sich in den letzten Jahren als wichtiges Mittel zur Steigerung von Softwarequalität bewährt. Denn agile Prozesse ermöglichen es, stets situationsgerecht, flexibel und angemessen zu handeln. Der Einsatz agiler Methoden verringert zudem nachweislich das Risiko des Scheiterns von Softwareentwicklungsprojekten.

Diese Popularität und der Erfolg agiler Methoden haben dazu geführt, dass insbesondere Scrum zunehmend auch bei der Softwareentwicklung im Maschinen-/Anlagenbau eingesetzt wird. Dabei erfolgt die Einführung nur selten auf der sogenannten grünen Wiese, sondern meistens in einer langjährigen Prozesslandschaft. Außerdem ist die Softwareentwicklung im Maschinen-/Anlagenbau durch ihren Bezug zu einer konkreten Maschine oder Anlage geprägt, woraus sich andere Rahmenbedingungen ergeben als bei klassischen Softwareprojekten. Zusätzlich wurde die Softwareentwicklung im Maschinen-/Anlagenbau meist unzureichend berücksichtigt, so dass bislang wenig spezifiziert wurde und auch kaum etablierte Prozesse vorhanden sind.

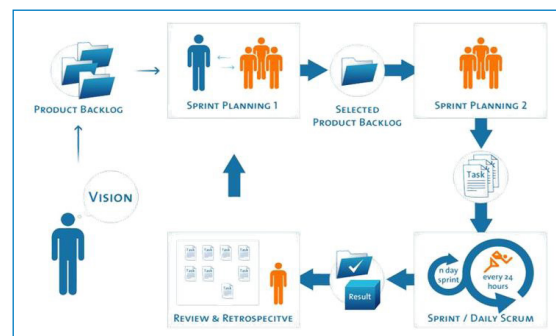
Wenn nun ein agiles Verfahren wie Scrum eingeführt werden soll, gehen damit große Chancen einher, jedoch birgt die Einführung auch Risiken, die minimiert bzw. eliminiert werden müssen.

Die Einführung agiler Entwicklungsmethoden zielt auf eine höhere Planungssicherheit und schnell-

lere Reaktionsmöglichkeit bei geänderten Rahmenbedingungen ab.

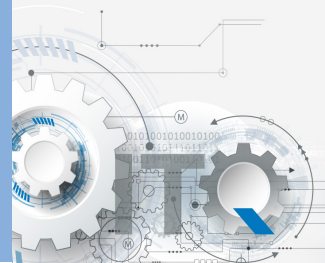
Scrum, der aktuell prominenteste Vertreter agiler Methoden, basiert auf drei Säulen:

- **Transparenz** des Projektfortschritts durch regelmäßige Kommunikation
- **Qualitätsprüfung** in kontinuierlichen Reviews
- **Anpassbarkeit** durch kurze Zyklen (z.B. 4 Wochen)



Ablauf eines Sprints

Nur durch diese strukturierte Reduktion der Komplexität können die geforderten Funktionsumfänge und insbesondere die damit einhergehenden nichtfunktionalen Aspekte, wie Safety und Security, beherrscht werden. Die von uns betreuten Kunden entwickeln ihre Softwareprojekte zielgerichtet und mit einer kontinuierlichen Optimierung der Prozess- und Produktqualität. Wir nutzen diese gewonnenen Erkenntnisse und haben sie bereits erfolgreich auf eine ganzheitliche, agile Entwicklung mechatronischer Systeme ausgeweitet.



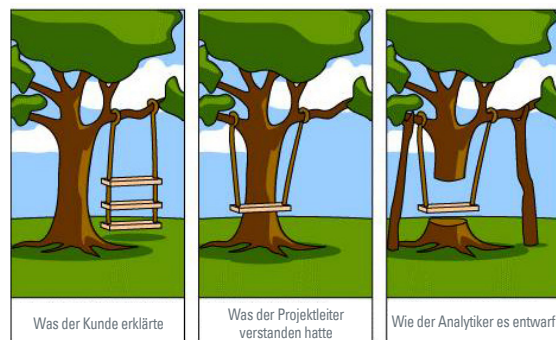
Anforderungsmanagement

Grundstein für erfolgreiches Engineering

Der Maschinen- und Anlagenbau wird mit zunehmenden Anforderungen aus dem Markt hinsichtlich individueller Maschinenkonfigurationen, maßgeschneiderter Konzepte und der Einbindung von Subsystemen konfrontiert. Nicht zuletzt werden durch Industrie 4.0 die Herausforderungen an die Entwicklung intelligenter und vernetzter Maschinen/Anlagen immer größer. Um der Flut von Anforderungen und der steigenden Komplexität Herr zu werden, muss ein professionelles Anforderungsmanagement existieren. Hierbei ist es wichtig, dass Anforderungen frühzeitig, detailliert und überprüfbar spezifiziert werden. Anforderungsänderungen müssen zudem dokumentiert werden, um während der Entwicklung agil darauf zu reagieren. Ansonsten besteht die Gefahr, dass die Projektkosten das geplante Budget signifikant übersteigen und wichtige Termine nicht eingehalten werden können.

Besonders im Anlagen- und Sondermaschinenbau gleicht keine Anlage/Maschine der anderen. Ein professionelles Anforderungsmanagement verlangt einen modernen Produktentstehungsprozess, in dem die Anforderungsklä rung verankert ist. Als Vorgehensmodell bietet sich das Quality-Gate-Modell oder das V-Modell an. Das Quality-Gate-Modell hat den Vorteil, dass alle drei wichtigen Disziplinen (Mechanik, Elektrik, Software) gleichzeitig betrachtet werden. In einem ersten Schritt ist es sinnvoll, eine Grundstrukturierung der zu betrachtenden Anlage/Maschine zu erarbeiten. Hierbei ist vor allem die

funktionale Strukturierung von Bedeutung. Ist der sogenannte „mechatronische Setzkasten“ erarbeitet, können die Anforderungen nun anhand der einzelnen Funktionen geklärt und definiert werden.



Um die Anforderungen über den gesamten Entwicklungsprozess zu verfolgen und ggf. anzupassen, ist ein übergeordnetes Tool nötig, welches zur Verwaltung und Überwachung aller Anforderungen vorgesehen ist. Beispiele hierfür sind Jira oder Redmine. Die beiden Tools unterstützen eine agile Entwicklung und bieten u.a. die Anbindung an Testmanagementsysteme.

Wir helfen Ihnen, ein professionelles Anforderungsmanagement zu erarbeiten und einzuführen. Dies geschieht immer anhand von konkreten Projekten oder Aufträgen. Auch die Betrachtung Ihres Entwicklungsprozesses und die anschließende Verankerung der Optimierungen in dem vorhandenen Prozess gehört zu unseren Leistungen. Zusätzlich helfen wir bei der Auswahl oder Anpassung der vorhandenen Tools für das Anforderungsmanagement.

Systemspezifikation

Klare Entwicklungsvorgaben durch Systemspezifikation

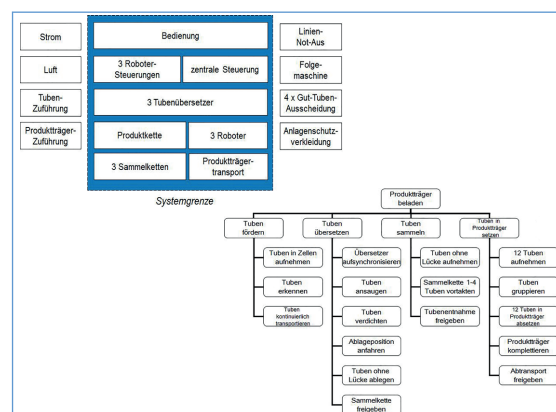
Mit dem Wandel von der reinen Mechanik hin zu der Mechatronik im Maschinenbau steigt auch die Bandbreite der Anforderungen und die Komplexität der Projekte. Damit einher geht die Notwendigkeit einer strukturierten Interdisziplinarität, die u.a. durch eine frühzeitige Klärung der Schnittstellen erreicht werden kann. Durch eine klare Entwicklungsvorgabe in Form einer ausführlichen Systemspezifikation können Aufwand und Kosten einer Entwicklung reduziert werden. Während der Erstellung der Spezifikation werden Anforderungen geklärt und dokumentiert. Des Weiteren wird durch die Systemspezifikation die Grundlage für eine zuverlässige Zeit- und Kostenplanung geschaffen.

Die Systemspezifikation ist eine sehr strukturierte Beschreibung mit dem Ziel, Merkmale und Ausprägungen zu definieren, zu quantifizieren und zu beschreiben. Diese Merkmale dienen als Grundlage zur Entwicklungsarbeit und können als Akzeptanzkriterien bei Abnahmen oder Übergaben verwendet werden. Die Systemspezifikation besteht aus folgenden Gliederungspunkten:

- Einführung
- Systembeschreibung
- Rahmenbedingungen
- Externe Schnittstellen

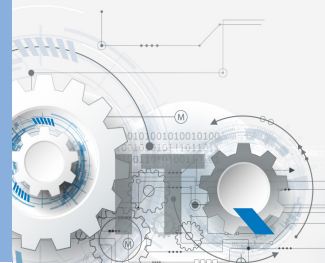
Die **Einführung** gibt einen kurzen Überblick, warum das System entwickelt wird, welche grobe Struktur es haben und wo es zur Anwendung kommen soll.

Die **Systembeschreibung** beinhaltet die vollständige Beschreibung des gesamten Systems. Neben der Umgebung, Konfiguration und der Betriebsarten werden auch die Funktionen und Eigenschaften des Systems beschrieben sowie mögliche Anwendungen und Abläufe, aus denen sich später Testszenarien ableiten lassen. Anschließend werden in **Rahmenbedingungen** alle Anforderungen wie mechanische, elektrische und Betriebsbedingungen beschrieben. Anforderungen an Normen, Projektentwicklung, Dokumentation, Checklisten gehören ebenso dazu. Die **externen Schnittstellen** detaillieren eindeutig alle Anforderungen an die Schnittstellen des Systems und Einbindung in das Gesamtsystem.



Systembild und Funktionsbaum

Wir helfen Ihnen dabei, die Systemspezifikation in Ihrem Unternehmen einzuführen und sie als zentrale Arbeitsgrundlage für Entwicklungen zu etablieren. In gezielten Workshops wird dabei in interdisziplinären Teams Ihre Vorlage für die Spezifikation erarbeitet.

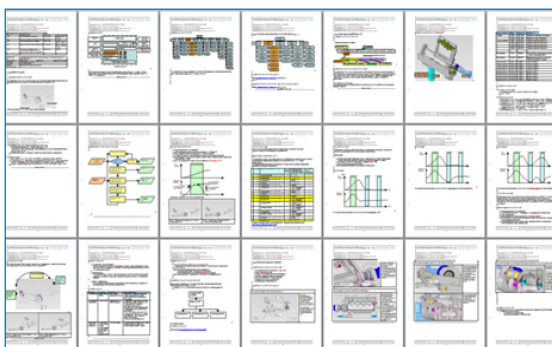


Funktionsbeschreibung

Steigerung der Effizienz im Engineering

Im heutigen Maschinen- und Anlagenbau ist der Anteil der Software sehr hoch, Tendenz stark steigend, so dass Maschinen/Anlagen inzwischen eher als mechatronische Systeme bezeichnet werden. Die Entwicklung solcher Systeme stellt die Entwicklerteams vor neue große Herausforderungen, da es vermehrt zu Wechselwirkungen zwischen den Disziplinen kommt. Um ein optimales Zusammenwirken, ein sogenanntes „Simultaneous Engineering“, zu erreichen und die entstandene Steigerung der Komplexität im disziplinübergreifenden Entwicklungsprozess erfolgreich zu bewältigen, müssen strukturierte Vorgehensweisen und Methoden etabliert werden.

Die Funktionsbeschreibung, eine möglichst detaillierte Beschreibung für neue Funktionen, ist der erste zentrale Schritt in der Spezifikation des mechatronischen Systems. Vor allem bei Projekten mit hohem Innovationsanteil ist eine Funktionsbeschreibung essentiell.



Beispiel einer Funktionsbeschreibung

Erste Schritte der Funktionsbeschreibung sind:

- Erstellen einer Übersicht des Gesamtsystems mit allen Schnittstellen zu anderen Systemen
- Einführung einer einheitlichen Terminologie über alle Disziplinen hinweg
- Modellierung einer Funktionshierarchie (Baumstruktur) aus elementaren Funktionen für das Gesamtsystem

Durch den Funktionsbaum wird eine komplexe Gesamtfunktion in weniger komplexe, handhabbare Teilfunktionen unterteilt. Im zweiten Schritt werden für die Lösungen der Teilfunktionen jeweils eigene Dokumente erstellt. Dies beinhaltet neben einem mechatronischen Aufbau und einer detaillierten Ablaufbeschreibung auch Punkte wie beispielsweise Betriebsarten, Einstellparameter, Störungen und Testszenarien.

Wir unterstützen Sie in Ihrem Entwicklungsprozess anhand bestehender oder anstehender Entwicklungen, bei der Erarbeitung der nötigen Inhalte und der Einführung von mechatronischen Funktionsbeschreibungen. Damit ist es Ihnen möglich, Ihre Spezifikationen in der Entwicklung eindeutig, strukturiert und frühzeitig gemeinsam für alle Disziplinen zu definieren. Auf dieser Grundlage sind Sie in der Lage, die Entwicklungszeiten zu verringern, da die unterschiedlichen Abteilungen früher mit ihrer Arbeit beginnen können und die Lösungen detailliert spezifiziert sind.

Software-Architektur

Mehr als nur Programmierung

Konstruktion von Software

Der Softwareanteil im Maschinenbau steigt stetig, jedoch ist das Image der Softwareentwicklung häufig noch davon geprägt, dass diese ohne große Aufwände schnell erledigt werden kann. Dabei wird die Tatsache ignoriert, dass Software analog zur Entwicklung mechanischer Komponenten konstruiert werden muss. Eine Software benötigt eine Architektur, die alle Anforderungen erfüllt und zukunftsfähig strukturiert ist. Bei der Konstruktion der Architektur gilt es folgende Fragen zu beantworten:

- Was ist die Systemumgebung der Software (Schnittstellen des Systems)?
- Welche funktionalen und nichtfunktionalen Anforderungen werden an die Software gestellt?
- Welche Aufgabe hat die Software?
- Welche (Software-)Technologien werden eingesetzt (Wiederverwendung)?

Die Planung und Dokumentation der Softwarearchitektur stellt die Weichen für eine erfolgreiche Entwicklung. Unsere Software-Architekten nutzen moderne Methoden und Werkzeuge zur Spezifikation wesentlicher Bestandteile der Architektur:

- Strukturübersicht der Software (statisch)
- Darstellung der Interaktion aller Komponenten (dynamisch)
- Funktionsstrukturbaum
- Abläufe und Zustandsautomaten

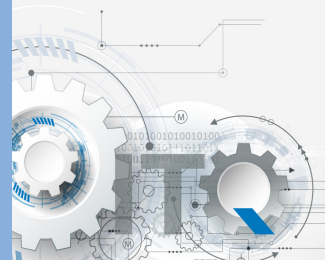
Analyse und Bewertung von Software

Viele aktuelle Softwarelösungen im Maschinenbau sind das Produkt einer kontinuierlichen und mehrjährigen Weiterentwicklung. Der Beginn dieser Entwicklung fiel oft in eine Zeit, in der der Softwareanteil gemessen an der Maschinenfunktion sehr gering und überschaubar war. Vor dem Hintergrund dieser gestiegenen Bedeutung der Software stellen sich viele Unternehmen folgende Fragen:

- Wie sieht unsere aktuelle SW-Struktur aus?
- Welche Qualität hat die aktuelle Software?
- Ist die Software „fit für die Zukunft“?

Um Softwarestruktur begreifbar zu machen und zu bewerten, muss diese visualisiert werden. ITQ erstellt auf Grundlage der Software-Visualisierung eine fundierte Bewertung der Strukturen und Zusammenhänge. Folgende zentrale Bestandteile sind das Ergebnis der Analyse und Bewertung der Software:

- Systembild, Komponenten-, Funktionsstruktur
- Visualisierung wesentlicher Abläufe
- Bewertung der Zukunftsfähigkeit und Komplexität der Architektur
- Design- und Technologiebewertung
- Bewertung der Code-Qualität
- Bewertung des Entwicklungsprozesses
- Handlungsempfehlungen und Entwicklungs-Roadmap



Modularisierung

Hohe Flexibilität bei geringen Engineering-Aufwänden

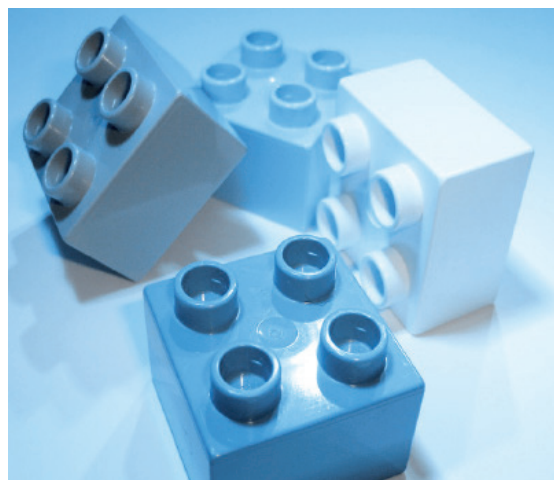
Die Verwendung modularer Systeme bzw. Plattformen für mechanische und elektronische Bauteile ist längst als Stand der Technik etabliert, um Engineeringaufwände zu begrenzen. Oftmals endet der Modul- bzw. Plattformgedanke jedoch an dieser Stelle.

Besonders im Anlagen- und Sondermaschinenbau gleicht keine Anlage bzw. Maschine der anderen. Ein baukastenorientiertes System verlangt aber nach wiederverwendbaren Modulen, die wahlweise um Optionen ergänzt werden können. Der erste Schritt zur Modularisierung einer Anlage ist die Analyse der bestehenden Systeme und der darin verbauten Betriebsmittel (Aktoren/Sensoren).

Die daraus resultierende Liste dient nicht nur für die Strukturierung und Erfassung der Anforderungen an die Funktion einer Maschine, sondern ist auch die Grundlage zur Standardisierung und Optimierung der Baugruppen. Ein modularer Aufbau sieht ein Grundgerüst mit Varianten vor. Demnach müssen in jeder Abteilung die bisherigen Arbeiten auf Gemeinsamkeiten/Unterschiede analysiert werden, um daraus in interdisziplinärer Zusammenarbeit Module zu definieren sowie zu strukturieren bzw. Varianten neu zu entwickeln.

Für die Module werden mechatronische Funktionsbeschreibungen erstellt, die die Kommunikation zwischen Vertrieb und Technik und auch innerhalb der technischen Disziplinen vereinfachen. Auf Grund-

lage des zu entwickelnden Varianten- und Optionskonzeptes müssen dann in den einzelnen Disziplinen die Engineering-Unterlagen teilweise restrukturiert bzw. neu entwickelt werden. Für jede Abteilung werden auf diese Weise Module definiert, welche dann als Bestandteile einer mechatronischen Bibliothek in ein Baukastensystem eingespeist werden können.



Auf Basis von bereits vorhandenen oder neu zu definierenden Einzelbaukästen kann sukzessive eine komplette mechatronische Strukturierung des Produktportfolios entwickelt werden.

Gerne unterstützen wir Sie beim Aufbau einer mechatronischen Bibliothek mit mechatronischen Modulbeschreibungen, wodurch Sie in der Lage sind, sehr schnell und flexibel auftragsspezifische Lösungen zu entwickeln.

HMI-Entwicklung

Moderne Bedieneroberfläche auf hohem Niveau

Eine moderne Bedieneroberfläche für Maschinen und Anlagen ist weit mehr als eine Softwarevariante des Start-Stop-Schalters. Die im Englischen als Human Machine Interface (HMI) bezeichnete Software ist für wesentliche Funktionen während Inbetriebnahme, Betrieb und Wartung verantwortlich. Oft verwaltet sie auch die Anbindung an Prozessleitsysteme und Datenbanken des Betreibers.

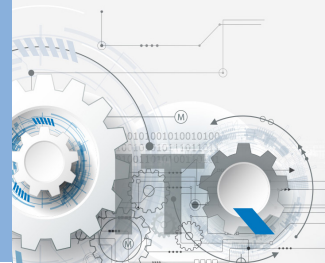


Das technologische Umfeld, in dem Bedieneroberflächen entwickelt werden, ist komplex. Werden Fehler in der HMI erst bei der Integration von Hard- und Software gefunden (Big Bang Integration), stehen einzelne Funktionen nicht sofort zur Verfügung. Verzögerungen von System-Integration und der Inbetriebnahme beim Kunden lassen sich kaum vermeiden. In der Software-Entwicklung entstanden in den vergangenen Jahrzehnten eine Vielzahl an Methoden, Vorschlägen und Best Practices, um Entwickler zu unterstützen, Programme auf hohem Niveau zu entwickeln:

- Modularer Aufbau einer Software-Architektur
- Prinzipien der Agilen Software-Entwicklung
- Prinzipien der fortlaufenden Integration
- Collective Code Ownership

Diese Methoden gilt es auf den Maschinen- und Anlagenbau anzupassen und in laufenden Projekten einzuführen. Durch eine modulare Software-Architektur mit klar getrennten Verantwortlichkeiten werden HMI-Applikationen aus einzelnen Bausteinen mit wenigen Abhängigkeiten untereinander konstruiert. Kritische Technologien können gekapselt und später ausgetauscht werden. Bei der Entwicklung von HMI-Projekten setzen wir etablierte Vorgehensmodelle wie Scrum, Extreme Programming (XP) und Software-Kanban ein. Regelmäßige Releases erlauben zuverlässige Aussagen über den Stand des Projektes und die übertragene Verantwortung motiviert die Entwickler zusätzlich.

In den Entwicklungsprojekten automatisieren wir die ständige Integration der Komponenten von Beginn an. Der Quell-Code wird auf einem zentralen Server verwaltet und nach jedem Commit kompiliert. Danach führt der Server Codeanalysen und Unit-Tests automatisiert aus, um die Qualität der Komponenten abzusichern. Etwa einmal täglich werden zusätzlich Integrations- und Oberflächentests ausgeführt und sichern die Funktionalität des Systems.



Software-Implementierung

Mit Struktur zum Projekterfolg

Moderne Software-Entwicklungsprojekte sind aufgrund der immer stärkeren Verzahnung von Software mit nahezu allen Komponenten der Maschine hochkomplex. Die daraus resultierende zwingend notwendige disziplinübergreifende Zusammenarbeit steigert die Komplexität der Entwicklungspro-

zesse zunehmend. Diese zu beherrschen fordert anwendungsgerechte Methoden, Prozesse und Werkzeuge. Wir programmieren Ihre Software-Applikationen für Sie und unterstützen Sie zudem bei der Auswahl neuer Werkzeuge und Methoden. Folgende Tools und Techniken wenden wir an:

Programmiersprachen (Software-Engineering):

- C/C++, C#, Java, VBA
- IEC 61131-3, Assembler
- HTML5, Javascript
- SQL, Stored Procedures

Software-Architekturen:

- OOA/OOD, UML, SysML
- Enterprise Architect, Microsoft Visio
- Distributed Systems

Entwicklungsumgebungen:

- Microsoft Visual Studio (2003 – 2013)
- Eclipse, Qt Creator
- WinCC

Steuerungen:

- Siemens (Step7, WinCC, Simotion Scout)
- B&R (Automation Studio)
- 3S (CoDeSys)
- Beckhoff (TwinCAT)
- Bachmann (Solution Center/MPLC)
- Rockwell (RsLogix)

Datenbanken:

- MySQL Administrator
- Microsoft SQL Server
- NoSQL
- Hibernate

Versionsverwaltung:

- CVS, Subversion (SVN)
- Perforce, Git

Test- und Qualitätssicherung:

- MSTest, NUnit, JUnit, Qt Test
- CppUnit, Jenkins
- LabView
- Squish, Coded UI
- Team Foundation Server

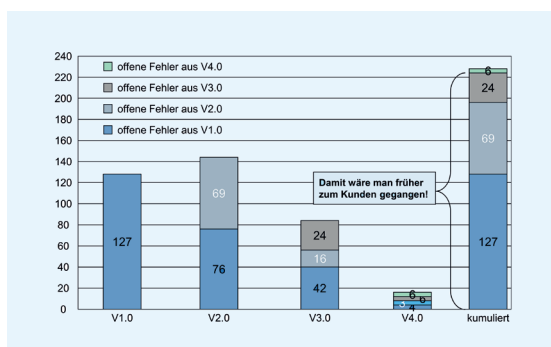
Simulationswerkzeuge:

- TrySim
- Emulate3D
- Industrial Physics
- ISG-virtuos
- Matlab Simulink

Testautomatisierung

Nachhaltige Qualitätssicherung bei höchsten Echtzeitanforderungen

Die lange Lebensdauer moderner Softwaresysteme verbunden mit ständiger Weiterentwicklung und Fehlerbehebungsmaßnahmen erfordert die wiederkehrende Durchführung von Testmaßnahmen. Diese Regressionstests haben zum Ziel, bereits bestehende Funktionen zu validieren und negative Wechselwirkungen durch vorgenommene Änderungen auszuschließen. Die zunehmende Individualisierung bis hin zur „Losgröße Eins“ erfordert, dass diese Tests mit möglichst geringem Aufwand durchführbar sind.



Fehlerreduktion durch systematisches Testen

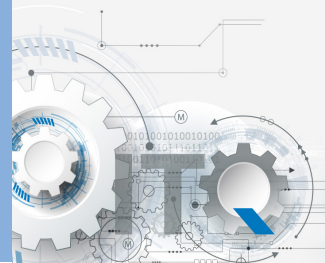
Die manuelle Durchführung der Regressionstests verursacht jedoch neben den Personalkosten auch hohe zeitliche Verzögerungen bis zur Rückmeldung des Testergebnisses. Daher sollten zumindest grundlegende Funktionen automatisiert getestet werden.

Um neue Funktionalität früh zu prüfen und vorhandene Fehler zu finden, muss in allen Phasen der Entwicklung getestet werden. Dies beginnt bei den einzelnen Bausteinen der Software und führt über

integrierte Subsysteme bis zum Gesamtsystem. Dazu werden aus den Funktionsbeschreibungen und Expertengesprächen die notwendigen Testfälle zur Abdeckung der geforderten Funktionalität abgeleitet. Auf Basis dieser Analyse erfolgt eine Strukturierung der Testfälle, um alle wesentlichen Performanz- und Schnittstellenanforderungen abzubilden. Idealerweise sollten alle Testfälle auf Knopfdruck durchführbar sein und nach kurzer Zeit ein schnell zu überblickendes Ergebnis liefern.

Anhand der beschriebenen Methode werden die Testfälle und das Testsystem definiert. Die für das Testsystem eingesetzten Werkzeuge reichen von Unit-Test-Frameworks für den Test einzelner Bausteine über Werkzeuge zum automatisierten Oberflächentest bis zu Simulationsumgebungen oder Teilaufbauten der Hardware für den Systemtest. Auch Werkzeuge wie Matlab und LabView können für die Programmierung von Testfällen oder zur Simulation der Testumgebung zum Einsatz kommen.

Die implementierten Tests werden dann von einem Continuous Integration System (vollständige Automatisierung der Softwareerstellung und Testdurchführung) ausgeführt und ein Testreport erstellt. Die Ausführung kann je nach Laufzeit der Tests nach jeder Änderung des Quell-Codes oder einmal täglich erfolgen. Schlägt ein Test fehl, so kann der Fehler vom Entwickler auf Knopfdruck reproduziert und schnell behoben werden.



Simulation

Integrierendes Werkzeug in der mechatronischen Entwicklung

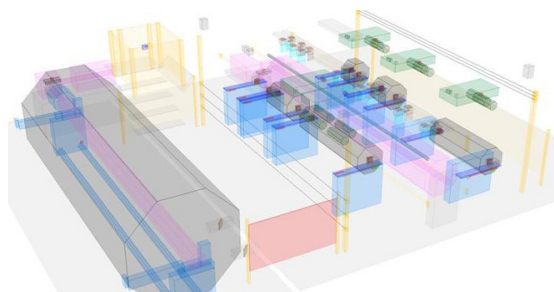
Der Einsatz moderner Simulationssysteme hat sich im Bereich der virtuellen Inbetriebnahme bereits durchgesetzt. Jedoch steckt die Methodik in anderen Bereichen des Entwicklungsprozesses in den Anfängen. Dabei werden gerade in den frühen Projektphasen die Weichen für eine erfolgreiche und kosteneffiziente Projektdurchführung gestellt, wozu angepasste Simulationsbaukästen einen wichtigen Beitrag leisten können.

Der Maschinen- und Anlagenbau wird mit zunehmenden Anforderungen aus dem Markt hinsichtlich individueller Maschinenkonfiguration, maßgeschneiderten Konzepten und der Einbindung von Subsystemen konfrontiert. Zusätzlich erwarten die Kunden deutlich reduzierte Lieferzeiten und eine schnelle Inbetriebnahme der Maschine/Anlage. Kurze Projektdurchlaufzeiten sind dabei entscheidend für eine erfolgreiche Platzierung auf dem Markt. Dabei gelingt es mit klassischen Methoden immer weniger, frühzeitig die Leistungsfähigkeit einer Anlagenkonfiguration oder von neu entwickelten Funktionskonzepten zu verifizieren.

Daher ist auch in frühen Projektphasen der Einsatz geeigneter Werkzeuge erforderlich, um die Qualität der erstellten Lösung abzusichern. Heutige Simulationswerkzeuge können dies in Verbindung mit einem entsprechend aufgebauten Modellbaukasten und einer guten Prozessintegration leisten. Im Simulationswerkzeug können Einzelkomponenten Ihrer

Maschine/Anlage entsprechend einem mechatronischen Modulkonzept nachgebildet werden. Auf dieser Basis können in der Projektierung kundenspezifische Maschinen- und Anlagenkonfigurationen mit wenig Aufwand erstellt werden. Durch den Einsatz eines funktionalen Demonstrators werden:

- neue Funktionsprinzipien schnell und prototypenhaft erstellt und getestet,
- verschiedene Konzepte einfach verglichen und Optimierungen durchgeführt,
- bereits in frühen Phasen Abstimmungen über Disziplinen ermöglicht.



Simulation eines Rollenschneiders

Das Modell findet anschließend in der Entwicklung als interdisziplinäres Abstimmungswerkzeug zwischen Mechanik-, Elektrik- und Software-Entwicklung Verwendung. Wir unterstützen Sie dabei, das zu Ihren Anforderungen am besten passende Simulationswerkzeug auszuwählen. Unsere breite Erfahrung aus einer Vielzahl durchgeführter Projekte mit unterschiedlichen Systemen und gutem Marktüberblick gewährleisten eine fundierte Entscheidung.

Benchmark von Methoden und Tools

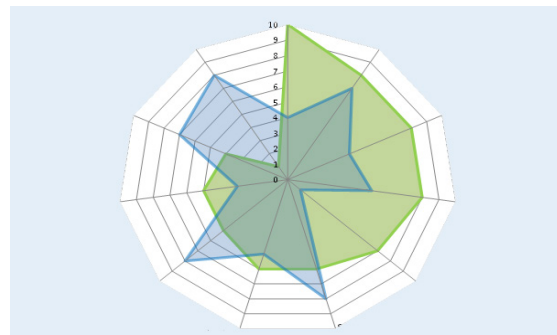
Effiziente Bewertung und Auswahl von Engineering-Methoden und -Tools

Innovative mechatronische Produkte erfordern ein effizientes interdisziplinäres Engineering. Voraussetzung hierfür sind ineinandergreifende Prozesse und durchgängige Werkzeuge. Die große Auswahl an existierenden Lösungen erschwert jedoch die Auswahl. Um zu nachhaltigen Entscheidungen über Methoden und Werkzeuge der nächsten Generation zu gelangen, müssen die benutzerspezifischen Anforderungen und Use Cases bekannt sein. Auf dieser Grundlage bietet ein Benchmark das Potenzial, Stärken und Schwächen ausgewählter Lösungen darzulegen.

Tools alleine helfen wenig, wenn der Prozess nicht stimmt. Die Prozesse aller an der Entwicklung Beteiligten müssen optimal ineinandergreifen, um kundenspezifische Produkte bei einem optimalen Kosten-Nutzen-Faktor anbieten zu können. Gleichzeitig müssen auch die zur Entwicklung benötigten Werkzeuge ideal aufeinander abgestimmt sein. Insellösungen und der damit verbundene Aufwand zur Pflege von Daten sind zu vermeiden. Denn ein durchgängiges Engineering hilft nicht nur Aufwand zu sparen.

Systematische Engineering-Benchmarks unterstützen nachhaltige Entscheidungen auf Grundlage klarer Anforderungen. Ausgehend vom Status Quo werden im Zuge der Vorbereitung die zentral zu betrachtenden Use Cases identifiziert und kundenspezifische Anforderungen erfasst und definiert. Auf

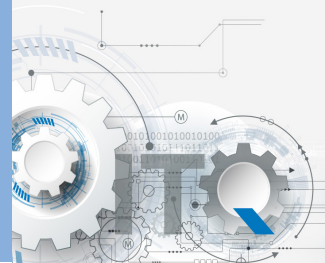
Grundlage einer klaren Bewertungsmetrik können dann heute etablierte Methoden und Werkzeuge systematisch identifiziert, ausgewählt und einem Benchmark unterzogen werden. Im Zuge einer Auswertung werden individuelle Stärken- und Schwächen-Profile der Lösungen dargelegt. Damit besteht die Möglichkeit des direkten Vergleichs anhand realer Use Cases. In Summe schaffen die Ergebnisse eine verlässliche Grundlage für eine nachhaltige Entscheidung bezüglich der Auswahl von Methoden und Werkzeugen im mechatronischen Engineering.



Benchmark eines Tools im Vergleich zu „Best-In-Class“

Die zentralen Bestandteile unserer Benchmarks:

- Dokumentierte Prozesse, Use Cases und Anforderungen als Ausgangspunkt
- Übersicht existierender Lösungen
- Detaillierte Ausarbeitung der Stärken und Schwächen
- Direkter Vergleich von Methoden/Werkzeugen
- Konkrete Handlungsempfehlungen



Forschungsprojekt NUCLEI

Aufbau von transnationalen Netzwerken

Zehn Partner aus sechs verschiedenen mitteleuropäischen Ländern haben sich für die nächsten drei Jahre zusammengetan mit dem Ziel, das überholte Innovationsmanagement Modell, basierend auf lokalen Technology-Scoutings zu ändern. Des Weiteren soll ein transnationaler Wissenspool geschaffen werden, welcher Innovationen im Bereich der Fertigungs- und Verarbeitungsindustrie grenzüberschreitend unterstützt.

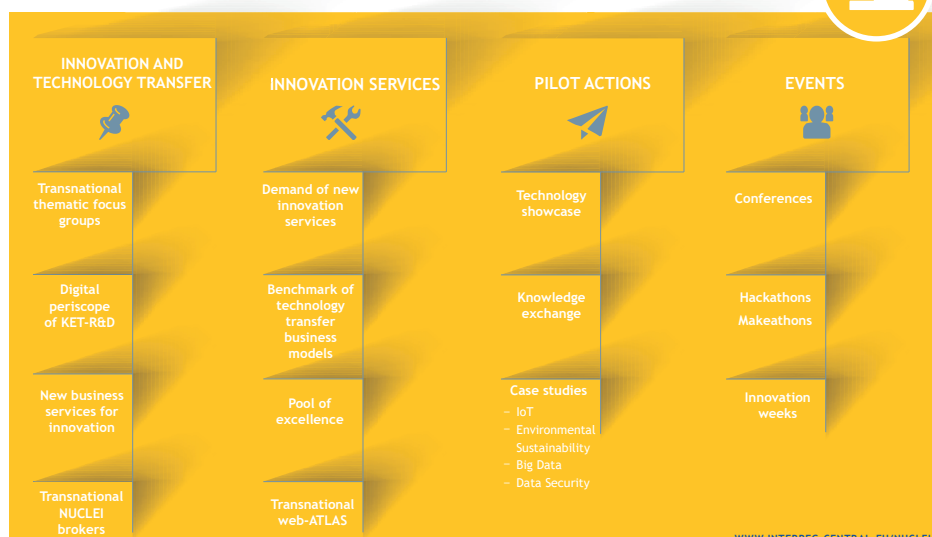
Das Akronym **NUCLEI** steht für **“Network of Technology Transfer Nodes for Enhanced Open Innovation in the Central European Advanced Manufacturing and Processing Industry”** (Aufbau von neuen Innovations-Modellen und Netzwerken für einen innovativen Technologie-Transfer zwischen den Mitteleuropäischen Industrie- und Wirtschaftsverbänden)

und wird mit 2,4 Millionen € von Interreg Central Europe gefördert. Die Herausforderung des NUCLEI-Projekts liegt darin, innovationsbezogene Dienstleistungen zu verbessern und effizientere, grenzüberschreitende Wertschöpfungsketten in der Automobil- und Elektroindustrie sowie auch in den Bereichen IT-, Robotik und mechanische Automatisierung zu fördern.

Der Austausch zu verschiedenen Aspekten und Ideen im Bereich Innovation wird vor allem helfen, existierende Technologietransfermodelle und Dienstleistungen für die Fertigungs- und Verarbeitungsindustrie zu charakterisieren. Technologietransfer ist ein kostspieliger Prozess, bei dem oft Doppelarbeit geleistet wird und Marktreifeanforderungen nicht zeitgerecht erfüllt werden können.



NUCLEI OBJECTIVES

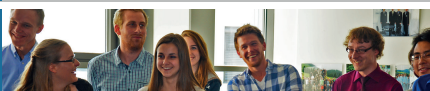


ITQ Akademie

Modulares Ausbildungskonzept

Um den Herausforderungen des Systems Engineering und den technologischen Anforderungen softwareintensiver mechatronischer Systeme gerecht zu werden, bedarf es geeigneter Strukturen, ausgefeilter Entwicklungsprozesse und motivierter, gut ausgebildeter Mitarbeiter.

Im Rahmen unserer ITQ Akademie unterstützen wir Unternehmen auf Basis eines modularen Schulungskonzeptes, das über verschiedene Hierarchiestufen und Aufgabenbereiche innerhalb des Unternehmens führt.

Modulares Ausbildungskonzept	
Management	 ■ Grundlagen und Bedeutung des Systems Engineering ■ Verständnis für mechatronische Projekte und Prozesse
Ingenieure	 ■ Besseres Verständnis für die Arbeit im interdisziplinären Team ■ Bessere Kenntnis im Umgang mit Software
Studenten	 ■ Erfahrung im Projektmanagement und Förderung der Soft Skills ■ Erweiterung des Software-Verständnisses
Schüler	 ■ Wirkzusammenhänge verstehen ■ Teamarbeit unterstützen und selbstständiges Denken fördern

Wir kümmern uns aber nicht darum, schon vorhandene Mitarbeiter geeignet auszubilden und zu schulen, sondern gehen ganz aktiv in Universitäten und Schulen, um dort direkt vor Ort dafür zu sorgen, dass die nachwachsende Generation sich für

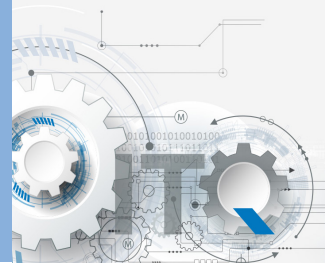
Technik begeistert und technische und soziale Wirkzusammenhänge frühzeitig erkennt.

Die einzelnen Seminare unseres modularen Ausbildungskonzeptes haben einen hohen Praxisbezug, der durch interaktive Übungen, Diskussionsmöglichkeiten und Feedbackrunden gekennzeichnet ist. Durch langjährige Projektarbeit, in der unsere Trainer sowohl beratend als auch operativ tätig sind, wird kompetentes Know-how weitergegeben.

Aktuell bieten wir Ihnen Workshops und Schulungen in folgenden Bereichen an:

- Mechatronische Entwicklungsmethodik
- Praxisnahe Softskills für mechatronische Projekte in Entwicklung und Produktion
- Agiles Engineering für Software
- Systematisches Testen von Software
- Simulationsmethodik in der Entwicklung
- Versionsmanagement von Software
- Security in der Industrie 4.0

Neben diesen technologisch und methodisch ausgerichteten Schulungen führen wir **Sensibilisierungs-Workshops für Management und Ingenieure** durch. Grundlage dieser Workshops ist der Technologie-Baukasten Lego Mindstorms, mit dem sehr anschaulich und für jeden nachvollziehbar in kleinen „Entwicklungs-Projekten“ die Bedürfnisse eines strukturierten Systems und Software Engineering dargestellt werden kann.



Showcase MI5

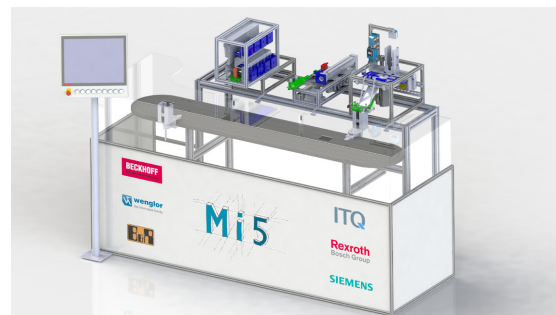
Engineering zum (Be-)greifen

In immer mehr Unternehmen reift die Erkenntnis, dass die Produktivität des Engineerings genauso wichtig ist wie die Produktivität in der Produktion. Denn nur Unternehmen, die fähig sind, komplexe, meist interdisziplinär gelagerte Produkte schnell und in guter Qualität zu engineeren, können langfristig international bestehen. Anders als in der Produktion ist es im Engineering jedoch nicht so leicht – rein durch „Anschauen“ der Produktionsstätten – die Arbeitsweise und Leistungsfähigkeit im Entwicklungsprozess zu „begreifen“ und adäquate Verbesserungsmöglichkeiten zu finden.

Um diese Hürde des „Begreifens“ zu verkleinern, bedarf es aus Sicht der ITQ GmbH sowie von Partnern aus Industrie, Forschung und Lehre, eingängiger und buchstäblich „greifbarer“ Anwendungsbeispiele. In diesem Sinne arbeitet ITQ gemeinsam mit Studenten der TU München, Hochschule München und TH Ingolstadt sowie internationalen Studenten der ETH Zürich und der La Salle Universität in Barcelona an dem Projekt MI5, das anhand eines begreifbaren Showcases darstellen soll, wie mechatronisches Engineering „idealerweise“ durchzuführen ist.

Gedankliche Grundlage für diesen Showcase ist eine neu zu entwickelnde Produktionslinie für einen großen Food-Konzern, mit dem ITQ eng zusammenarbeitet. Die skizzierte „Engineering-Demonstrationsanlage“ soll sowohl virtuell als auch real aufgebaut werden, um auf mehreren Abstraktionsebenen

darstellen zu können, wie die einzelnen Schritte des Engineerings konkret durchzuführen sind. Dabei sollen alle erforderlichen Unterlagen (Anforderungs-, Lösungs- und Test-Spezifikationen) möglichst idealtypisch erstellt werden. Die Umsetzung der Gesamtanlage soll in definierten Inkrementen erfolgen. Zur Absicherung der Ergebnisse dieser Einzelschritte sollen unter Zugrundelegung eines iterativen, agilen Prozesses, insbesondere in frühen Phasen des Projektes, Simulationen durchgeführt werden.



Skizze der Produktionsanlage

Die Ergebnisse des stufenweise umzusetzenden Projektes, beispielsweise Dokumente, Konzepte, Vorgehensweisen, Hard- und Softwareprototypen, sollen einem breiten Publikum vorgestellt werden, um zu zeigen, dass sauberes mechatronisches Engineering nicht nur in der grauen Theorie möglich ist. Zur Erstellung dieses „Engineering-Showcases“ wird das Team von namhaften Unternehmen der Zuliefer-Industrie (Automation, Engineering-Toolketten) wie Beckhoff Automation, Bosch Rexroth, B&R sowie Siemens unterstützt.

MI5 steht für:

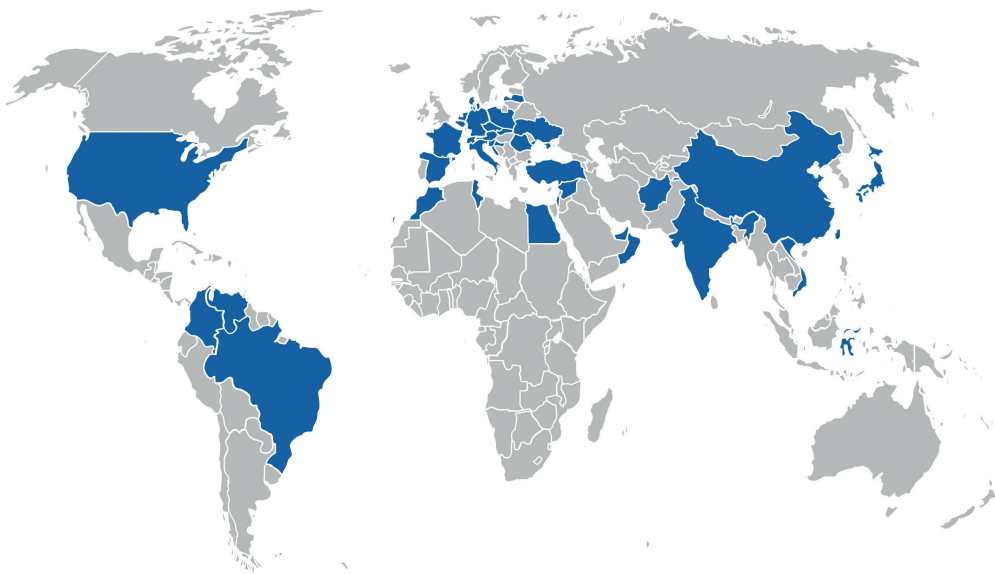
M = Mechatronik Engineering

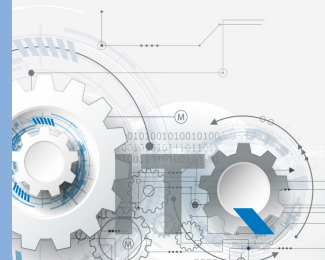
I5 = idealtypisches Engineering:

- innovativ
- interdisziplinär
- international
- inkrementell
- iterativ

Arbeiten in Netzwerken

Wir reden nicht über Netzwerke, sondern wir bauen aktiv internationale Netzwerke bestehend aus führenden Vertretern von Industrie, Hochschulen und Verbänden.





Unsere Basis

Das Team

Unser junges Team setzt sich aus kompetenten, praxiserfahrenen und hilfsbereiten Persönlichkeiten zusammen. Für uns ist Humor und Spaß am Job ebenso wichtig wie die erfolgreiche Umsetzung unserer Projekte und Aufgaben.

Wie wir das schaffen? Durch Kompetenz und Engagement. Wir denken analytisch und strategisch, schätzen die Herausforderung von technischen Innovationen und pflegen einen regen interdisziplinären Austausch. Denn die Grundlage für ein gutes Team ist Kommunikation auf Augenhöhe und ein adäquater, sozialer Umgang miteinander.

Dies ist die Voraussetzung, um die Anforderungen der vierten industriellen Revolution zu meistern, denn: Nur derjenige, der den sozialen Herausforderungen innerhalb eines Teams gerecht wird, kann am Ende erfolgreich sein. Dies impliziert einen Managementansatz bei dem nicht Strukturen und Hierarchien, sondern Aufgaben im Fokus stehen – dies gilt bei ITQ seit den Unternehmensanfängen vor 20 Jahren. Und dafür benötigen wir Industrie 4.0 fähige Menschen mit einer hohen Sozialkompetenz – den Social Technical Networker – der neben dem fachlichen Know-how ein hohes Maß an Kommunikationsfähigkeit, Flexibilität und Motivation mitbringt.



Portfolio

Mechatronisches Systems Engineering

- Mechatronische Organisationsberatung
- Benchmark von Technologien und Tools
- Auswahl von Engineeringwerkzeugen
- Projektmanagement
- Prozessoptimierung und -verbesserung
- Interimsmanagement
- Mechatronische Modularisierung
- Analyse von vorhandenen Systemen
- Definition von Varianten und Optionen
- Spezifikation von Systemen und Software
- Durchführung von Machbarkeitsstudien
- Entwicklung von Prototypen

Software Engineering

- Agile Entwicklung von Software
- HMI-Entwicklung/Mobile Devices
- SPS-Programmierung für alle Hersteller
- Software-Restrukturierung
- Automatische Softwareanalyse
- Entwurf einer geeigneten Architektur
- Implementierung produktreifer Applikationen
- Automatisiertes Testen
- Entwicklung und Implementierung von Test-Managementsystemen
- Virtuelle Inbetriebnahme
- Auswahl geeigneter Simulationswerkzeuge und vollständige Modellerstellung
- Spezifikation und Durchführung von Testfällen

Mechatronische Ausbildung

- Agile Entwicklung
- Erstellung modularer Systeme
- Systematisches Testen
- Test von Software/Systemen (SIL/HIL)
- Mechatronisches Engineering
- Team-Management/Team-Building
- Präsentation und Rhetorik
- Management Sensibilisierungs-Workshops