competence in mechatronics

software and systems engineering

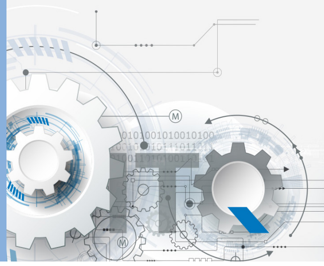
ITQ

Vision or Truth?

Software with Machine or Machine with Software

„The man with a new idea is a crank until the idea succeeds“

Mark Twain



Thinking in Systems

Software with Machine: Today More Than Ever

When we co-founded the Software Section within the German Engineering Association more than 15 years ago to meet the increasing significance of the subject, the association's statutes had to be changed to allow companies with software background to join the organization.

But since that time, the basic truth of the provocative maxim **„software with machine“** would replace **„machine with software“**, has become consensus. In the age of Industrie 4.0, when software with machine is called „Cyber-Physical System“, this is particularly obviously.

This technological change has strongly influenced the character of (Systems) Engineering and software is not simply programmed anymore – as in the past – but developed systematically.

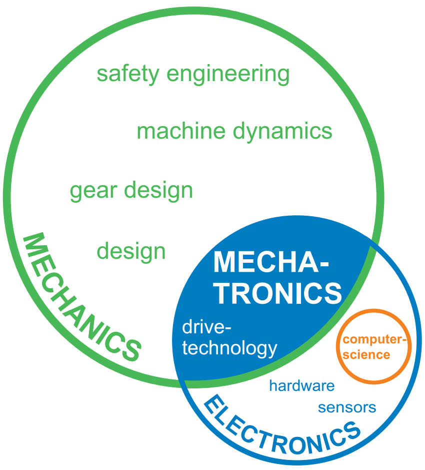
The paradigm change towards ever more software functions will not slow down in the foreseeable future. To really master this change, professional education in academia and industry must be practice-oriented and adapted to the actual technological requirements.

The three spheres of activity

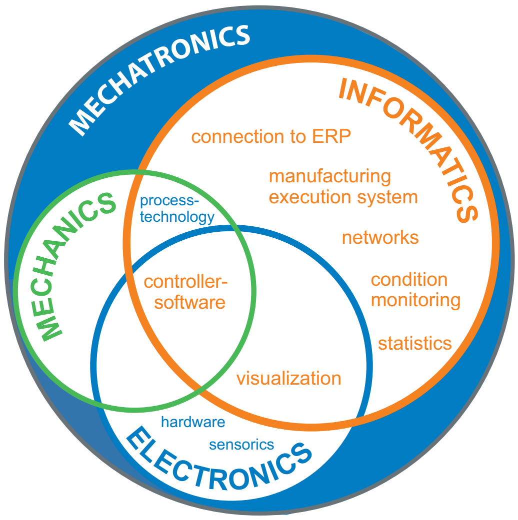
- mechatronic systems engineering
- agile/structured software engineering
- mechatronic education for all age groups

constitute ITQ's cornerstones and will be described in detail hereafter.

MECHATRONICS
A SUB-ORDINATE TASK



MECHATRONICS
AN INTEGRATED APPROACH



Our Spheres of Activity

More than Programming

Several of our customers lack the development capacities necessary to fulfill all market requirements. So we support our customers in their software implementation in many cases.

When we execute an implementation project, we often notice that the software requirements are not really clear. For that reason we support not only in the implementation phase but also during the earlier phases of analysis and software design as well as with software quality control. Clear specification of requirements and modular software design are important preconditions of systematic quality control.

But we not only support our customers in software engineering, but also see them through the introduction of mechatronic systems engineering. In these projects we advise the customers on how to adapt the organization and which technologies and methods are best in the given situation.

The subject of mechatronic training is highly significant for us. There is a great shortage of engineers in Germany, which is why we not only address the question of educating students as efficiently as possible but also offer a multitude of activities to inspire and train young and old in technical context.

Software Engineering

- Analysis
- Design
- Implementation
- Test and Quality Control
- Software
  - Control (HMI, PLC, Embedded)
  - Processing (PLM, MES, Remote Maintenance)

Systems Engineering

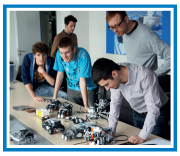
- Mechatronic organization consulting
- Benchmarking of technologies and tools
- Mechatronic modularization
- Project management
- Mechatronic sytems engineering
  - Projects in time, on budget
  - Speeding up commissioning

Mechatronic Education

Management

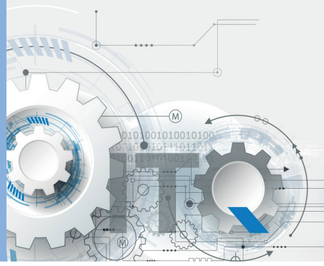
Engineers

Students



Pupils

Children



Modus Operandi

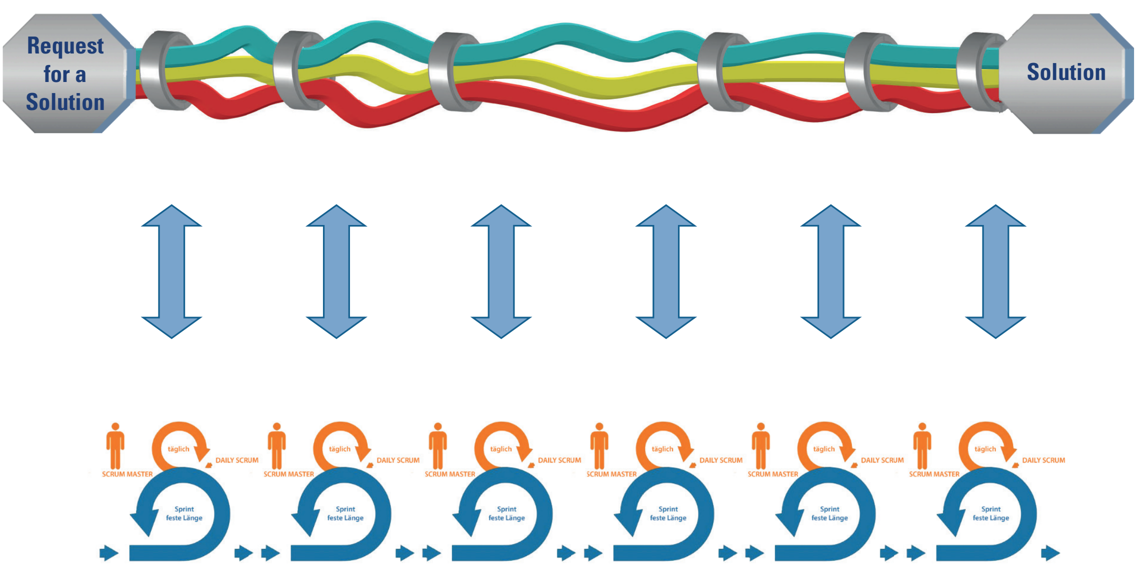
Ordered but still agile

The central idea of the mechatronic development technique favored by us is the Quality Gate model, as recommended by the German Engineering Association VDMA, which describes the elements of a „healthy interdisciplinary development process".

In order to increase the efficiency of development processes, all parts of a project are integrated into a continuous procedure based on established project management techniques. The goal of the Quality Gates recurring within the development process is to continuously synchronize the results of the involved disciplines and therefore secure them based on concrete product requirements.

As with agile approaches to the development of complex software systems, projects are divided into a sequence of „sprints". The results of the sprints are reviewed and assessed in interdisciplinary workshops at each Quality Gate. The result of the assessment – visualized by a simple traffic-light color code – determines the requirements and goals of the next iteration. For the reviews clear-cut check lists are defined.

Using this combination of a stringent look-ahead Quality Gate approach and an agile and iterative development methodology, transparency and flexibility are achieved even in complex projects.



Status Quo

Traditional, Sequential Engineering

Today in many companies engineering is still done in rather traditional way. At the start of a project, external customers or internal project managers usually formulate the requirements in a quite rough and fuzzy manner.

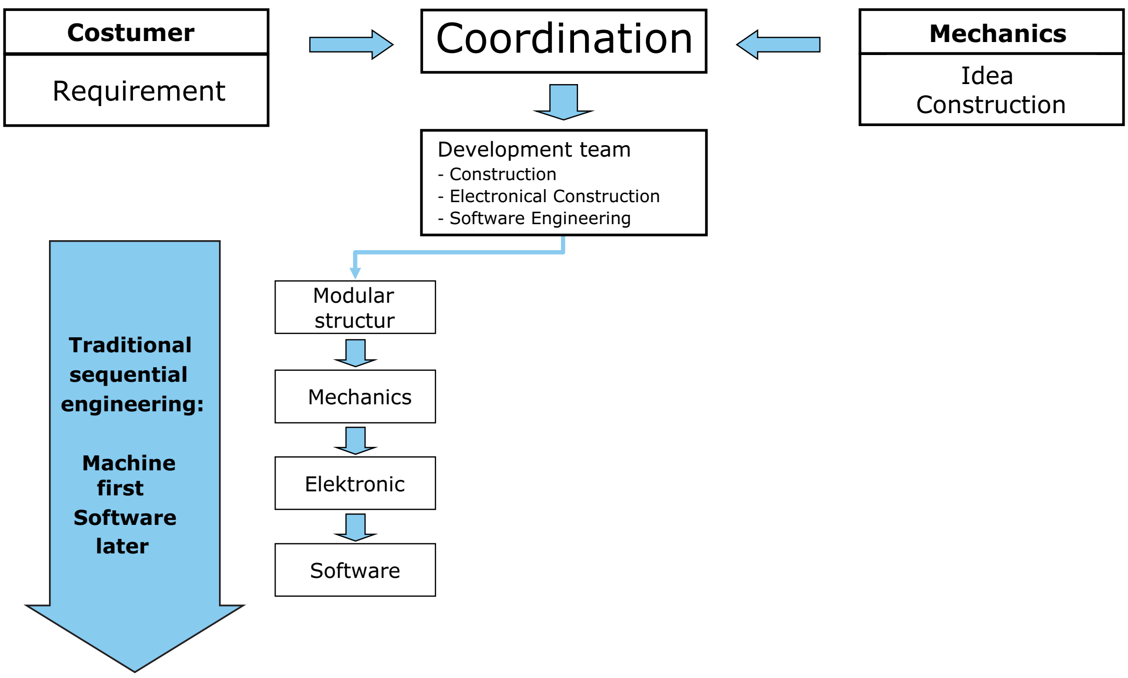
Then the requirements are discussed with (sales) staff of the company who frequently are or used to be constructing engineers. Often, first ideas for constructive solutions are discussed in this coordination phase.

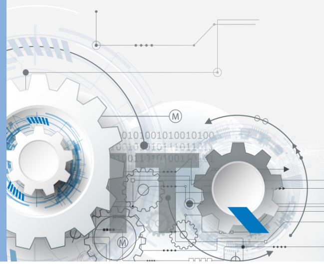
The implementation team usually consists of mechanical, electrical, and software engineers. The foundation for their work is the list of components. The mechanical constructions are done first, then the electrical ones. Usually the software implementation begins rather late in the progress of the project.

The workflow of this traditional way of engineering can be summarized like this:

„The machine is complete, now only the software is missing".

As we all know, this sequential way of engineering comes with some drawbacks, especially in a time when the software content of a machine continuously increases.





The Goal

Systems Engineering Coordinated between Disciplines

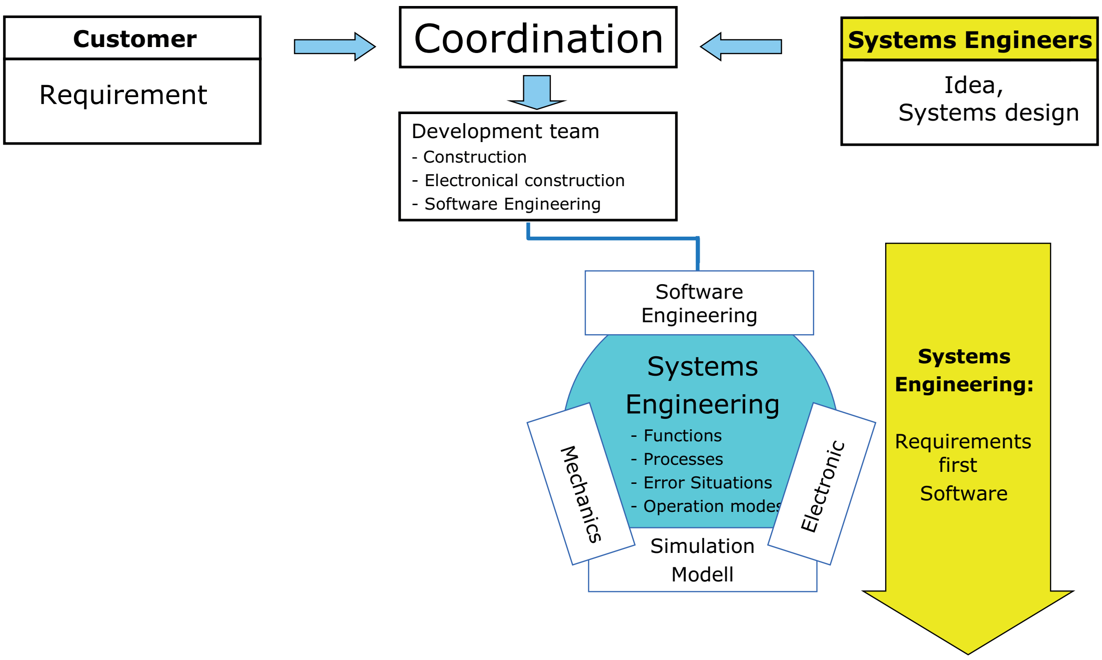
This is why we have to reflect together on the future shape of engineering that is coordinated across all disciplines.

To achieve significant improvement of the process, so-called "systems engineers" not only need to think about constructive concepts in the starting phase of a project, but also about the functionality of the whole system including software.

So the functions and processes of the system have to be defined in the early phases of the project. Furthermore the possible errors and the desired reactions of the system have to be determined. In addition the functions and processes available in a given operating mode must be defined. As soon as these technical requirements are settled, the first software tasks can be tackled. To check the proper functioning of the software, simulation models are deployed. Only when the preliminary work is completed, mechanical operations should start.

This way of mechatronic systems engineering can be summarized in the formula: **First resolve the functional requirements (in detail), then start software implementation early.**

This new way of engineering requires not only new processes in the companies but also agile but structured procedures.



Agile Development Methodology

Act Quicker and More Flexibly

Agile methods have proved to be an important instrument to increase the software quality because agile processes make it possible to react always in an appropriate, flexible, and adequate way. Besides, the usage of agile methods demonstrably minimizes the risk of failure of the software development project.

The popularity and success of agile methods have resulted in Scrum being adopted more and more in software development for mechanical engineering, too.

It is rarely introduced from scratch but rather added to a long-standing process landscape. Besides, software development in mechanical engineering is affected by intimate connection with a concrete machine or plant, leading to other general requirements than in classic software projects.

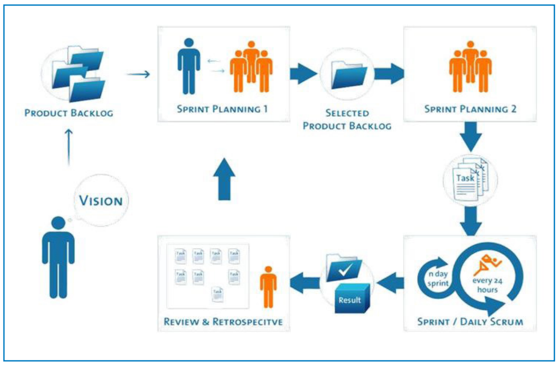
Also, since software development has previously been insufficiently considered in mechanical engineering, there hasn't been a lot of specification and established processes.

Introducing an agile process such as Scrum involves great opportunities but also risks that must be minimized or eliminated.

The implementation of agile development methods aims at higher planning reliability and quicker reaction to changed constraints.

Scrum, the currently most prominent exponent of agile methods, is based on three pillars:

- **Transparency** of project status by regular communication
- **Quality check** in continuous reviews
- **Adaptability** by short cycles (e.g. 4 weeks)



Sprint Workflow

The required functional range and especially the associated non-functional aspects, such as safety and security, can only be mastered by this structured reduction of complexity.

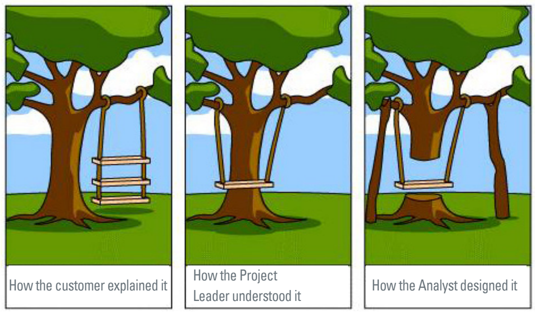
The customers mentored by us develop their software projects in a goal-oriented way and continuously optimize the quality of process and product. We harness the knowledge gained by this and have already expanded it to a holistic, agile development of mechatronic systems.

Requirement Management

Foundation for Successful Engineering

Mechanical engineering is increasingly confronted with new market requirements such as individual machine configurations, customized concepts and embedding of subsystems. Thats why Industrie 4.0 aggravates the challenges the development of intelligent and interconnected machines or plants presents. To handle the flood of requirements and mounting complexity, a professional requirement management is needed. An important point is that requirements are specified early, in detail, and in a verifiable way. Also, changes in requirements during development must be documented to allow agile reaction. Otherwise there is a danger of project budget cost overrun and important deadlines not being kept.

Especially in plant and special purpose machinery manufacture, no plant or machine is like another. A professional requirement management demands a modern product creation process with integrated requirement clarification. As process models, the Quality Gate model or the V model can be used. The Quality Gate model has the advantage of considering all three important disciplines (mechanical engineering, electrical engineering, and software) at the same time. The first step is developing a structural framework concept for the plant/ machine. The most important aspect here is the functional structure. As soon as the so-called „mechatronic type case" is finished, the requirements can be clarified and defined on the basis of the particular functions.



Copyright 2005 Paragon Innovations, Inc.

In order to track the requirements during the whole development process and possibly match them to new conditions, an overriding tool for administrating and monitoring is necessary. Two examples are Jira and Redmine. Both facilitate agile development and offer connection to test management systems

We help you compile and introduce a professional requirement management in the context of actual projects or commissions.

Examination of your development process followed by integration of optimizations into the existing process is another of our services.

Additionally, we help you choose and adapt existing tools for requirement management.

System Specification

Clear Development Targets by System Specification

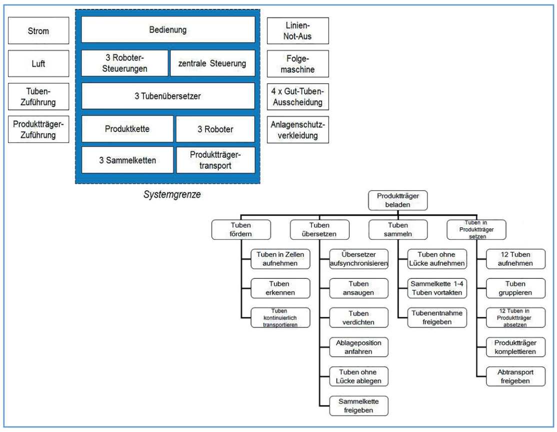
The change from pure mechanics to mechatronics in mechanical engineering gives rise to a broader spectrum of requirements and higher project complexity. It also necessitates structured multidisciplinarity, which can be achieved i.a. by early clarification of interfaces. By clear development targets in terms of a detailed system specification, effort and cost of development can be reduced. In the process of specification, requirements are clarified and documented. Furthermore the system specification is the fundament of a reliable time and cost management.

The system specification is a very structured description aiming at defining, quantifying, and describing characteristics and peculiarities. These characteristics are the foundation for development and can be used as acceptance criteria at delivery or commissioning. The system specification consists of:

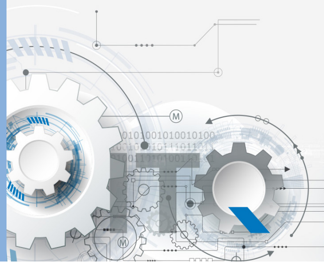
- Introduction
- System description
- Framework conditions
- External interfaces

The **introduction** shortly summarizes the reason for developing the system, its approximate structure, and its scope of application. The **system description** contains the complete description of the whole system. Besides the environment, configuration, and operation modes, functions and properties of the system are depicted, as well as possible applications and workflows from which test scenarios can be derived later. In the **framework conditions**, all requirements such as mechanical, electrical, and operating conditions are described. Requirements for standards, project process, documentation, and check lists are also part of this. The **external interfaces** uniquely determine all requirements for the system interfaces and the integration into the overall system.

We help you introduce system specification in your company and establish it as the central foundation for development. Interdisciplinary teams compile your specification template in specific workshops.



System structure and function tree



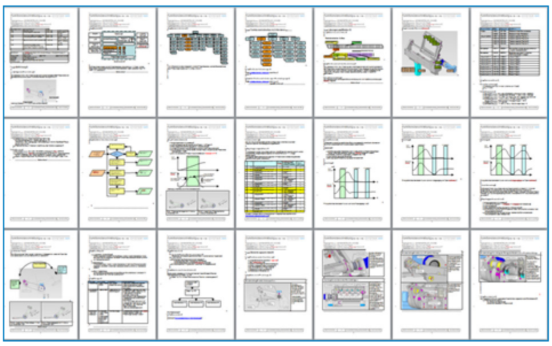
Functional Description

Efficiency Increase in Engineering

Today, the software content in mechanical engineering is very high with upward tendency, and machines and plants are described as mechatronic systems by now. The development of such systems confronts engineering teams with new and great challenges as interactions between the disciplines increase.

Structured procedures and methods have to be established in order to achieve optimal cooperation, a so-called „Simultaneous Engineering“, as well as to master the resulting complexity rise in an interdisciplinary development process.

The function description, an account of new functions in as much detail as possible, is the first central step of the specification of the mechatronic system. It is essential especially in projects with a high proportion of innovation.



Example for a function description

The first steps of the function description are:

- Compilation of an overview of the complete system with all interfaces to other systems
- Adoption of a consistent cross-discipline terminology
- Modelling of a function hierarchy (tree structure) of elementary functions for the complete system

The function tree divides a complex overall function into less complex manageable sub functions. In a second step, a special document is compiled for the solution of each sub function.

Besides a mechatronic composition and a detailed workflow description, items such as operation modes, setting parameters, faults, and test scenarios are involved.

We support you in your development process, within existing or upcoming projects, in drafting the contents and in launching mechatronic function descriptions.

With this you can define your specifications for development early in an unambiguous and structured way for all disciplines. Based on this you can reduce the development times as the various departments can start work earlier and the solutions are specified in detail.

Software Architecture

More than just Programming

Construction of Software

The software proportion in mechanical engineering increases steadily, but the image of software development is still often characterized by the idea that it can be done quickly with little effort. The fact that software has to be constructed like mechanical components is ignored. Software needs an architecture that fulfills all requirements and is structured in a sustainable way. When constructing the architecture, the following questions have to be answered:

- What system environment does the software have (interfaces of the system)?
- Which functional and non-functional requirements must the software fulfill?
- What is the function of the software?
- Which software technologies are used?

Planning and documentation of the software architecture sets the course for a successful development. Our software architects use modern methods and tools for specifying substantial parts of the architecture:

- Structural overview of the software (static)
- Delineation of the interaction of all components (dynamic)
- Function structure tree
- Workflows and state machines

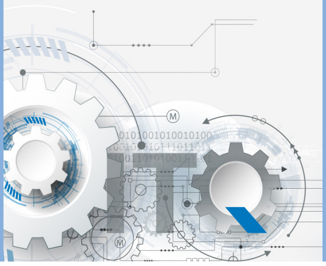
Analysis and assessment of software

Many current software solutions in mechanical engineering are the product of continuous and perennial enhancements. The development often started at a time when the proportion of software was tiny und easily understandable compared to the function of the complete machine. In light of the increased importance of software, many companies ask themselves the following questions:

- What does the software structure look like?
- What quality does the current software have?
- Is the software sustainable?

To make software structure comprehensible and assess it, it has to be visualized. ITQ does a well-founded assessment of structures and interrelations based on the software visualization. The following central components are the result of the analysis and assessment of the software:

- System image, component / function structure
- Visualization of essential workflows
- Assessment of sustainability and complexity of the architecture
- Assessment of design and technology
- Assessment of code quality
- Assessment of development process
- Recommended courses of action and development roadmap



Modularization

High Flexibility with Little Engineering Effort

The usage of modular systems or platforms for mechanical and electronic components has long been established as state of the art for limiting engineering effort. But often the module or platform concept ends at this point.

Especially in plant engineering and special purpose machinery manufacture, no plant or machine is like any other. But a building-block-oriented system demands reusable modules that can be supplemented by options. The first step of modularizing a plant is the analysis of existing systems and of in those systems installed equipment (actors/sensors).

The list resulting from this serves as a foundation not only of structuring and collecting the requirements for the function of a machine but also for standardization and optimization of components.

Modular structure implies a base frame with variants. Therefore, in each department, the previous tasks have to be broken down into similarities and differences in order to define and to structure modules in interdisciplinary cooperation as well as develop new variants.

For the modules mechatronic function descriptions are made which simplify the communication between sales and engineering and within the technical disciplines. Based on the variant and option concept to be developed, the engineering documents in each discipline have to be partly restructured or even developed new. For each department, modules are defined in this fashion which can be fed into a building-block system as parts of a mechatronic library.

The complete product portfolio can be successively structured mechatronically by defining new construction kits.

We gladly support you in constructing a mechatronic library of mechatronic module descriptions, allowing you to develop order-specific solutions quickly and flexibly.

HMI Development

Modern High-Level User Interface

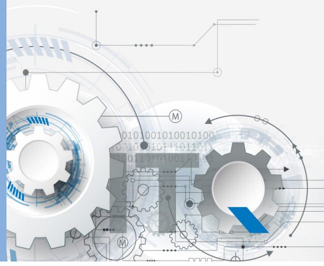
A modern user interface for machines and plants is far more than a software version of the start/stop switch. The software called Human-Machine Interface (HMI) is responsible for essential functions during commissioning, operation and maintenance. It also often manages the connection to the operating company's process control systems and databases.

The technological environment in which HMIs are developed is complex. Finding bugs in the HMI only during integration of hardware and software (Big Bang integration) may make individual functions temporarily unavailable. Delays of system integration and commissioning at the customer can hardly be avoided. In recent decades a multitude of methods, recommendations, and best practices arose in software development which are suitable for helping developers write high-level programs:

- Modular structure of software architecture
- Principles of agile software development
- Principles of continuous integration
- Collective code ownership

These methods must be adapted to mechanical engineering and introduced in ongoing projects. With a modular software architecture in which responsibilities are clearly separated, HMI applications are constructed from constituent building blocks with few interdependencies. Critical technologies can be encapsulated and later exchanged if necessary. When developing HMI projects, we apply established process models such as Scrum, Extreme Programming (XP) and Software-Kanban. Regular releases allow reliable statements about the state of a project, and conferring responsibility to developers enhances their motivation.

In the development projects we automate the continuous integration of the components from the beginning. The source code is administered on a central server and compiled after each commit action. Then, the server automatically performs code analysis and unit tests to ensure the quality of the components. About once per day, additional integration and user interface tests are executed to ensure the quality of the whole system.



Software Implementation

Project Success by Structure

Modern software development projects are highly complex because of the increasing interlocking of software with almost every component of the machine. The resulting necessity of cooperation across all disciplines increasingly enhances the complexity of development processes. To master these, methods, processes, and tools suitable to the application are needed. We program your software applications for you and support you in choosing new tools and methods. We apply the following tools and technologies:

Programming languages (software engineering):

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

Software architectures:

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

Development environments:

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

Programmable controllers:

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

Databases:

- MySQL Administrator
- Microsoft SQL Server
- NoSQL
- Hibernate

Version management:

- CVS, Subversion (SVN)
- Perforce, Git

Test and quality control:

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

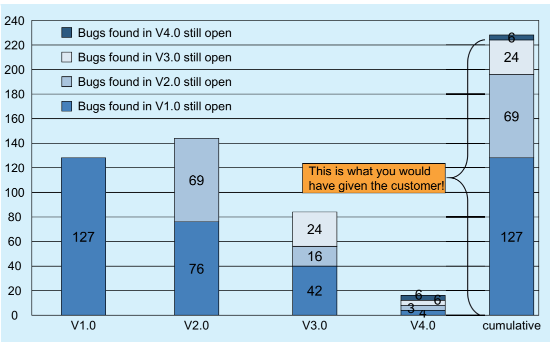
Simulation tools:

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

Test Automation

Sustainable Quality Control at Highest Real-Time Requirements

The long life of modern software systems, together with regular upgrades and bug fixes, requires recurring tests. These regression tests aim at validating already existing functions preventing negative interactions by changes in the code. The increasing individualization to the point of „lot size is one“, demands that these tests can be executed with as little effort as possible.



Bug reduction by systematic tests

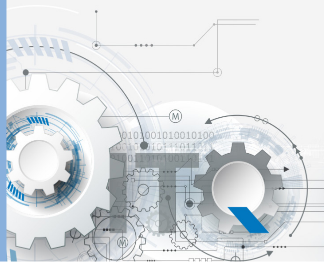
The manual execution of regression tests causes not only personnel costs but also a considerable time lag until the test result is known. Therefore at least basic functions should be tested automatically.

In order to check new functionality as early as possible and find existing bugs, tests are necessary in each development phase. This begins at single building blocks of the software and continues via integrated subsystems to the whole system. To achieve this the test cases for coverage of the functionality are deduced from function descriptions and expert meetings. Based on this analysis, the test cases are structured to map all the essential performance and interface requirements.

Ideally all test cases should be executable by button click and deliver an easy breakdown after little time.

Using the method described, the test cases and the test system are defined. The tools applied in the test system range from unit test frameworks for testing individual building blocks via tools for automated user interface tests to simulation environments or partial hardware systems for system test. Tools such as Matlab and LabView can also be used for programming test cases or simulating the test environment.

The implemented tests are then executed by a continuous integration system (complete automation of software compilation and test execution) and a test report is generated. The execution can occur, depending on the time needed by a test run, either after every source code change or once daily. If a test fails, the developer can reproduce the failure by pressing a button and quickly fix it.



Simulation

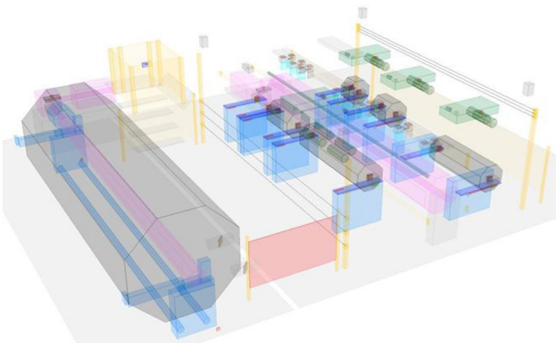
Integrating Tool in Mechatronic Development

The use of modern simulation systems has already gain acceptance in the area of virtual commissioning. But in other sectors of the development process, the methodology is still in the early stages. But especially in the early project phases, the course is set for a successful and cost-efficient project execution, and adapted simulation kits can contribute significantly to that.

Mechanical engineering companies are confronted by the market with increasing requirements regarding individual machine configuration, tailor-made concepts, and embedding of subsystems. Additionally the customers expect notably reduced delivery times and quick commissioning of the machine or plant. Short project processing times are a decisive factor for a successful market position. But with classical methods an early verification of a plant configuration or of newly developed function concepts is getting less and less probable.

For that reason, even in early project phases, the use of suitable tools is required to ensure the quality of the developed solution. Today's simulation tools can do this, in conjunction with an appropriately assembled model kit and good process integration. In a simulation tool, individual components of your machine/plant can be emulated according to a mechatronic module concept. Based on this, customer-specific machine and plant configurations can be created during project planning with little effort. A functional demonstrator will achieve:

- Quick prototypic tests of new functional principles
- Easy comparisons and optimizations of different concepts
- Coordination between disciplines in early phases



Simulation of a slitter winder

After that, the model is used in development as an interdisciplinary coordination tool between mechanical, electrical and software engineering. We help you to choose the simulation tool matching your requirements best. Our wide-ranging experience from a multitude of projects with diverse systems and good overview of the market ensure a well-founded decision.

Benchmarking of Methods and Tools

Efficient Assessment and Choice of Engineering Methods and Tools

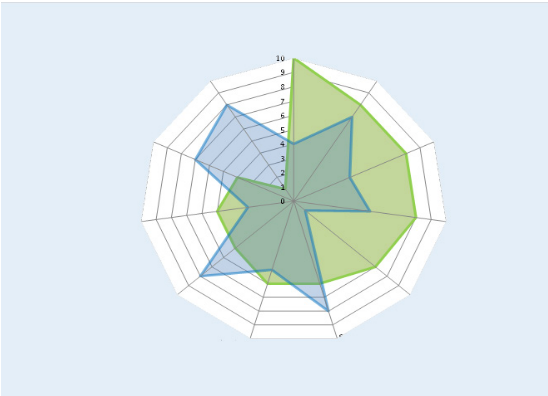
Innovative mechatronic products require efficient interdisciplinary engineering. The precondition for this is intermeshing processes and uniform tools. But the wide range of existing solutions complicates the choice.

To end up with sustainable decisions about methods and tools for the next generation, the user-specific requirements and und Use Cases must be known. On this basis a benchmark possesses the potential to expose strengths and weaknesses of selected solutions.

Tools alone are not sufficient enough if the process is wrong. The processes of all parties involved in development have to mesh optimally in order to offer customer-specific products with an optimum cost-value ratio.

At the same time all tools needed for the development must be coordinated ideally with each other. Isolated applications and the associated data maintenance effort have to be avoided. Because consistent engineering does not only help minimizing effort.

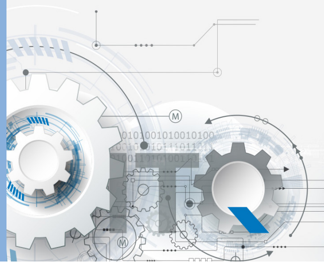
Systematic engineering benchmarks support sustainable decisions based on clear requirements. Starting at the status quo, the central Use Cases are identified and customer-specific requirements gathered and defined in the course of the preparation. Based on a clear assessment metric, established methods and tools can be identified, chosen and benchmarked systematically. In the assessment, individual profiles with strengths and weaknesses of the solutions are presented. With this, a direct comparison on the basis of real Use Cases is possible. In total, the results create a reliable foundation for a sustainable choice of methods and tools in mechatronic engineering.



Benchmark of a tool by comparison with another

Central components of our Benchmarks:

- Documented processes, Use Cases and requirements as starting point
- Overview of existing solutions
- Detailed elaboration of strengths and weaknesses
- Direct comparison of methods/tools
- Concrete recommendation



Research project NUCLEI

Building Networks - Invest in the Future

The European Union's most important automation and mechatronic industries are located in Central European regions. While their business dimension is intercontinental, their innovation-related services often remain local, thus slowing the transfer of research and development results into the industrial system. As a result, tech-transfer is an expensive process that often duplicates efforts and does not respond to the time-to-market requirements.

The challenge is to improve the innovation-related services to accelerate the transposition of key enabling technologies (KET) from EU-funded research & Central European labs into new end-user components and apps for the advanced manufacturing industries.

Therefore NUCLEI aims to change the obsolete innovation management model from a "local-based" technology scouting approach to a transnational pool of knowledge that supports advanced manufacturing innovation beyond regional borders. That is the reason why ten partners led by the Italian CRIT will work together in the next years to change this kind of management model.

„NUCLEI - Network of Technology Transfer Nodes for Enhanced open Innovation in the Central Europe advanced manufacturing and processing industry."

This will increase economic interdependencies among seven regions and encourage more effective transnational value chains in automotive, electrical industry, IT sector, robotic and mechanic automation. Such a joint knowledge sourcing approach will help the seven NUCLEI industrial clusters and its end-beneficiaries (corporations, SMEs, R&D performers) to generate new consortia/business deals for executing bold technological, product, and market projects.

ITQ Academy

Modular Training Concept

To meet the procedural requirements of systems engineering and the technological ones of software-intensive mechatronic systems, suitable structures, sophisticated development processes and motivated, well-trained staff is needed.

Within the framework of our ITQ Academy, we assist companies based on a modular training concept pervading several hierarchy levels and spheres of activities within the corporation.

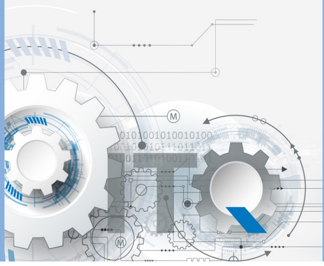
However, we not only train existing staff but actively address universities and schools in order to make sure on site that the upcoming generation gets enthusiastic about technic and catches technical and social correlations early on.

The individual seminars of our modular training concept are highly practice-relevant, indicated by interactive exercises, chances for discussions, and feedback rounds. Since our trainers have been consulting as well as developing for many years in many projects, competent know-how is being passed on.

Currently, we offer workshops and trainings in the following areas:

- Mechatronic development methodology
- Practically relevant soft-skills for mechatronic projects in development and production
- Agile engineering for software
- Systematic testing of software
- Simulation methodology in development
- Version management of software
- Security in Industrie 4.0

Besides these technologically and methodically oriented trainings, we conduct **awareness workshops for management and engineers.** The foundation of these workshops is the Lego Mindstorms technology kit, which can illustrate software engineering and the needs of a structured system in a very vivid, easily understood way in little „development projects".



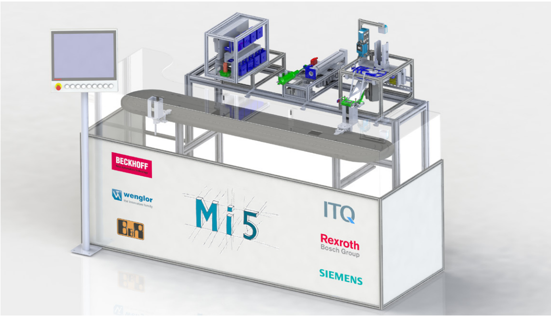
Mission Mi5

Mechatronical ideal Engineering

Productivity in engineering is as important as in production. But grasping procedure and productivity in the development process or finding adequate improvement opportunities only by taking a look at the facilities, is not as easy in engineering as in production.

As ITQ GmbH and partners from industry, research, and education see it, literally "graspable" examples of use are needed. In this spirit, ITQ, together with national and international students (Italy, Spain, Tunisia, China), are working on the Mi5 project, which demonstrates ideal mechatronic engineering using a comprehensive showcase.

The model project Mi5 puts into practice new educational concepts in the age of Industry 4.0. The goal is to implement innovative solution approaches and present ideal-typical engineering processes.



Sketch of the production device

Mi5 stands for:
M = Mechatronic Engineering
i5 = ideal engineering

- innovative
- interdisciplinary
- international
- incremental
- iterative

Innovative:

- Connecting control units from different companies in one system using OPC-UA connection.
- Rapid decrease of integration effort of new modules
- Real time processing of sensor data
- Developing new engineering procedures as well as tools to support them
- Using a full simulation as important tool to develop mechanics, electronics and software in parallel

Interdisciplinary:

- Appliance of approaches and methodology from different disciplinary fields
- Challenges in Informatics, Mechanics, Physics, Electrics and Math

International:

- Cooperation with different universities around Europe
- Student teams at Munich, Gran Canaria, Barcelona
- Students from all around the world

Incremental

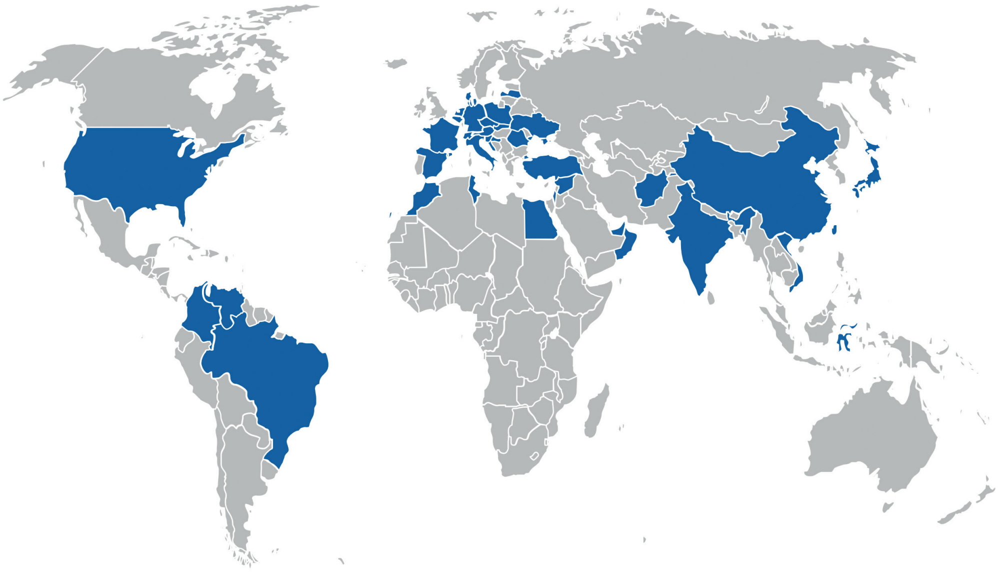
- Breaking the system down into small pieces
- Proceeding in small defined steps
- Using and improving different methods like Sprint planning, Scrum, and so on.

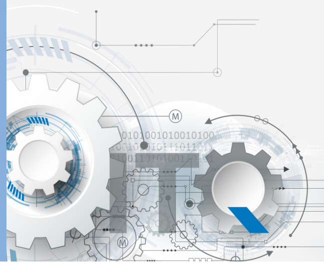
Iterative:

- Improving the system step by step
- Review at the end of each sprint to check if the incremental steps were sufficiently done

Working in Networks

We do not only talk about networks, we actively build international networks consisting of leading representatives of industry, academies and associations.





Our Basis

The Team

Our young team consists of competent, cooperative personalities with field experience. For us, humor and enjoying our work are as important as successful implementation of our projects and tasks.

How do we achieve this? The answer is: Through competence and commitment. We think analytically and strategically, we appreciate the challenge of technical innovations and we actively exchange knowledge between disciplines. The foundation of a good team are a high quality of communication and reasonable social terms. That is the precondition for mastering the requirements of the fourth industrial revolution, as only those who cope with the social challenges within a team can be successful in the end.

This implies a management approach where tasks are being focused instead of structures and hierarchies – this has been true for ITQ since the founding period 20 years ago.

Therefore we need Industrie 4.0 compliant people with a high social competence – "Social Technical Networkers" – who complement the expertise with a high degree of flexibility, motivation, and an ability to communicate.

Portfolio

Software Engineering

- Agile software development
- HMI development/mobile devices
- PLC programming for all manufacturers
- Software restructuring
- Automatic software analysis
- Design of suitable architecture
- Implementation of mature applications

- Automatic testing
- Development and implementation of test management systems
- Virtual commissioning
- Choice of suitable simulation tools and creation of complete models
- Specification and execution of test cases

Systems Engineering

- Mechatronic organization consulting
- Benchmarking of technologies and tools
- Choice of engineering tools
- Project management
- Process optimization and improvement
- Interim management

- Mechatronic modularization
- Analysis of existing systems
- Definition of variants and options
- Specification of systems and software
- Feasibility studies
- Development of prototypes

Mechatronic Education

- Agile development
- Generation of modular systems
- Systematic testing
- Testing of software/systems (SIL/HIL)

- Mechatronic engineering
- Team management/team building
- Presentation and rhetoric
- Management awareness workshops

ITQ GmbH ■ Parkring 4 ■ D-85748 Garching bei München ■ Phone: +49 89 321981-70
■ Fax: + 49 89 321981-89 ■ info@itq.de ■ www.itq.de