ARCHITECTURE-CENTRIC SOFTWARE AND SYSTEMS ENGINEERING
Excellent Software and Systems: Architecture-Centric Engineering

Many organizations struggle with the increasing but inherent complexity of their software systems. At the same time, they have to provide more and more functionality and react flexibly to changing requirements. Furthermore, software is often no longer developed by a single team but involves interacting organizations with large teams at multiple sites. Additionally, modern systems do not operate on their own but are strongly interconnected, resulting in software ecosystems. All these factors – if not managed well – bear the risk of entering the market too late, with insufficient quality, or even both.

Architecture is the key factor for success in the development and evolution of software systems. Architecture offers abstractions on the key properties, structure, and behavior of software systems and thus imposes order on the inherent complexity. Architecture also connects software systems to development organizations, processes, and other systems.

Effective architecting helps you to plan the functional and quality properties of a system in a predictable way. You can control the realization at the code level and keep systems maintainable. This allows managers to fulfill management constraints like time and budget. Consequently, architecture is the means for making the right decisions at the business and the technology level.

We consider architecture as the hub in software engineering. On the one hand, it connects business aspects with technology and balances the interests of all stakeholders of a system and their communication. On the other hand, it ties all activities of software engineering closely together, spanning all lifecycle phases of software systems such as development, operation, usage, and retirement.

The realization of excellent software systems requires the use of architecture as the central vehicle for prediction, analysis, and control.

We don’t see architecture as an end in itself. Rather, we are convinced that architecture requires explicit scenarios for its usages. Consequently, we tailor all activities related to architecture to the specific engineering context for maximum effectiveness.

Get the Maximum from Your Software by Using Architecture

Reap the benefits for your software and maximize your engineering success by exploiting the power of architecture in collaboration with Fraunhofer IESE.

Architect Your Success!

We offer you the knowledge and support

- to make fact-based decisions on the business, product, and technology levels
- to create high-quality architectures that adequately fulfill your requirements
- to significantly improve key quality properties of your software systems
- to effectively connect all artifacts and activities in software development related to the architecture
- to keep control over the quality of your software systems along the lifecycle
- to improve your organization’s architecture capabilities.

On the following pages we will show you how to benefit from great architectures and high architecture capabilities.

Sincerely,

Thorsten Keuler               Jens Knodel               Matthias Naab
In software and systems development, organizations often rely on components or services from sub-contractors, which allows them to focus on their core business. Such sub-contractors might be suppliers for certain control units in the automotive domain or companies delivering large parts of information systems in the airline industry. Parts acquired from other organizations mostly require tight integration with an organization’s own software systems. However, this integration often entails significant problems as systems might not be built for integration or only partly so. A common reason for this are mismatches of the architectures of the systems being integrated.

Many factors come into play when choosing one sub-contractor or another. One of the most decisive but often neglected factors is the system’s architecture. We help you to get a sound understanding of the product under consideration and we base this understanding on facts that only the architecture can deliver. To this end, we conduct neutral, objective, in-depth analyses and assessments of your software systems, clearly targeted to answer your concrete questions.

Besides the selection of sub-contractors, there are many other business decisions that are closely connected to architectural aspects of the related software systems. For many decisions, the risks call for a sound analysis involving the architecture. We offer industry-proven evaluation approaches and tools providing solutions to questions like: “Can we realize this key next generation feature with our current application landscape?”, “What do we need to provide and maintain multiple individual customizations of our product?”, “How does our product support the new paradigm of service orientation?”

IESE also helps you to build your next generation software systems. Clarify your business goals and requirements from the perspectives of all stakeholders and operationalize them into product requirements that are deliberately balanced by architecture decisions. Additionally, we can show you what new paradigms such as SOA might mean for you, what is only a hype topic, and where you can really benefit.
In industry, one of the key challenges is to design software systems that fulfill quality requirements such as performance, flexibility, or security. Wrong design decisions made at the architectural level might be recognized too late to be recoverable at low cost: For instance, hardware dependencies discovered during software production probably lead to costly rework across the entire development lifecycle. If issues are detected even later, such as performance issues in the field, the resulting costs are orders of magnitude higher.

For complex software systems, architectures need to be designed incrementally as an evolving set of coherent quality-specific solutions which are then systematically synthesized in order to address the quality requirements imposed on the product. In order to make architectural designs valuable, however, architectures need to comprise appropriate architectural models that allow reasoning about a product’s quality properties before it is actually implemented and tested.

We provide tool-based methodical guidance in architecture design and modeling. In other words, we help you to answer questions like “How should we structure the core functionality so that we can easily port it to a new hardware platform?” or “How responsive is the system if we use Publish-Subscribe as the main communication paradigm?” In order to deliver satisfying answers, we help to create consistent architectural models at the appropriate level of detail, keeping track of direct and indirect impacts revealed among solutions that address different concerns.

Eventually, architectures need to be documented in such a way that all relevant stakeholders are supported by the information conveyed by the architectural document. Therefore, the documentation needs to be understandable, focused on the respective stakeholder’s concerns, and always up to date. We help you create this documentation with a clear connection to the stakeholder concerns on the one hand and with a clear link from the model to the actual product artifacts (such as requirements specifications, source code, or test cases) on the other hand. Keeping the architectural model and the product artifacts in synch is one of the key success factors in this regard.
Today, the ever-increasing pace of technological progress is important for both the development and the actual deployment and operation of software-intensive products. The reasons for using a specific set of technologies are manifold in practice: In case of new product developments the technologies might be the result of management decisions, for instance based on observations of current technological trends. In the context of legacy systems, companies might be stuck with technologies due to the enormous risk of migrating critical systems to new technologies.

Whatever the reason: In order to take advantage of architecture-centric engineering, architects need to understand the consequences of a particular architectural decision for potential technologies and vice versa. Architecture evaluations with a focus on technologies allow fact-based reasoning and decision making in an objective manner. The results of technology prototyping in the context of architecture evaluations provide a solid basis for technology selection, while creating awareness of the pros and cons of the technology under investigation at the same time.

In this context, we help you to deal with non-trivial questions when it comes to selecting or rolling out development tools such as configuration management systems or integrated development environments. We also assist you in building architectural prototypes with selected technologies (e.g., databases or networking devices like bus systems or routers) and help you to make technology-related decisions based on facts.

In an ideal architecture-centric development environment, there are processes in place that deal with technologies and their lifecycles. First, there is a need for continuous awareness of cutting-edge technologies, of their capabilities and their drawbacks. This technological awareness is essential for enabling innovation. Secondly, there needs to be a systematic process that supports and controls the roll-out of new technologies in the developing company. Finally, the retirement of technologies must follow a systematic, planned process that is in accordance with the product structures affected by specific technologies. Here we help you to achieve a smooth transition to such processes by setting up and communicating relevant processes to the affected resources in combination with tailored on-the-job trainings.
A development organization cannot afford to step back and make a fresh start with a new architecture built from scratch: New products and new releases with new features have to be delivered constantly and existing products have to be maintained – resulting in a high workload for the organization’s experts. But those experts would also be required to provide valuable input and crucial domain knowledge for the design of software architectures.

With our competence in architecting and our experience gained from many technology transfer projects, we offer a solution to this dilemma. We complement your internal resources to build up your expertise in architecting:

• We bring in our neutral, technology-independent expertise from research.
• We tailor architecting to the needs of your development organization.
• We design innovative concepts and define solutions.
• We deliver proofs-of-concept and architectural prototypes.
• We promote architectural knowledge in your organization.

For a limited period of time, we support your development organization. We incrementally transfer methods, processes, and techniques until the internal team of architects can take over.

Effective architecting requires having a good start aligned with ongoing development. We help to carefully weigh strategic investments versus operative needs on a daily basis. We understand these challenges faced by our partners acting under tight pressure in evolution and migration.

Team up with the IESE experts in architecting and let us solve your evolution challenge. Regardless of whether you are replacing an integral technology, designing the next generation platform, or defining innovative solutions for extensible and flexible implementations, we can manage your evolution!
Architecture is the manifestation of a set of activities – the architecture practices, which define how the architecture is constructed, how design decisions will be propagated, and how the outcome of downstream activities will be reflected back into the architecture. Development organizations have a lot of options in terms of where to invest. And, of course, such optimization activities have to achieve a return on investment.

So investing into a practice where some degree of saturation has been reached does not appear to make sense. But how can a development organization figure out where it stands?

Being at the leading edge of research in software architecture and actively advancing the state of the art, IESE is able to assess your architecture practices. In contrast to purely process-oriented assessments, we explicitly emphasize our assessment on the output – project-specific ratings of the results of applied practices in combination with a rating of the level of application of the practices themselves.

Improvement of architecture practices is a worthwhile endeavor but requires clear objectives and priorities. Our experience-based approach to optimizing architecture practices is the Quality Improvement Paradigm (QIP). The underlying idea of QIP is to first set measurable goals to improve a given situation. Then we analyze to which degree the goals have been achieved and to which degree they have, in fact, improved the respective practice. Finally, experiences and lessons learned are packaged for later use. QIP enables continuous improvement with step-wise control and calibration.

Optimizing architecture practices calls for clear insights on the maturity of the architecture practices at your organization. As an objective assessor, we can lift your architecture practices to the next level and sustain optimizations over time!
The architecture of a software system defines the most important design decisions and is responsible for appropriate realizations of all types of requirements under specific constraints, such as technical, organizational, or legal constraints. Consequently, we consider architecture as the hub of software engineering.

That is, architecture is considered the mediator between

- business and technology: it connects the requirements of any kind of business and a technical solution in a software system.
- different organizations dealing with software systems, such as organizations using a system, building a system, or operating a system.
- different stakeholders of a system by balancing their requirements and facilitating their communication.
- key requirements like cost, time, and quality, which require sound balancing.
- different lifecycle phases of a software system: development, usage, operation, and retirement.

Our software and systems engineering approach is architecture-centric: This allows architecture to assume its mediating and balancing role. Architecture is not only a phase in software development; rather, it is a continuous activity throughout the whole lifecycle of a software system, connecting all other software development activities. This enables us to develop and maintain excellent software systems with predictable quality and time.

Despite the central role of architecture, it must never become an end in itself. Therefore, we make the usages of architecture explicit and adequately tailor all architecture activities to the concrete context at hand in order to keep development effective and efficient.

All our competencies in architecture are packaged in our product ACES: Architecture-Centric Engineering Solutions. It is the basis for our projects, where we solve our customers’ challenges with a combination of services.
ARCHITECTURE-CENTRIC ENGINEERING SOLUTIONS: IESE’S COMPETENCIES

Supporting IESE customers in solving their challenges requires continuous evolution of our key competencies. All results from research in architecture as well as our experiences from projects with industrial customers are packaged in our product ACES. Internally, our competencies in ACES are organized as follows:

**Architecture Lifecycle Management** packages our competencies in how to manage the architecture during and beyond development projects. Further more, it covers key aspects of how architecture is integrated into software engineering.

**Architecture Significant Requirements** packages our competencies in the area where architecture meets requirements. We can systematically elicit and organize requirements that are significant for the architecture. In particular, all types of stakeholders are involved and business goals and quality requirements are captured in architecture scenarios.

**Architecture Decomposition Framework** packages our competence in how to describe architectures with relevant views. Therefore, we elaborated a generic matrix, which serves as a basis for individual usage in describing a system. For example, the matrix distinguishes among aspects of the system in its context, the system at runtime, and the system during development.

**Design, Modeling, and Migration** packages all our competencies regarding how architecture design is done. This includes methods and guidelines for making design decisions in system decomposition, in achievement of quality attributes, modeling design decisions, and pinpointing everything in a useful documentation.

**Decision Propagation and Reflection** packages our competencies regarding how to prepare and distribute architectural information in order to support development activities like development or testing. Architectural decisions have to be propagated in order to be consistently realized and information gained during realization has to be reflected back to the architecture.

**Rapid Architecture Evaluation** packages our competencies in architecture evaluation. This covers evaluating the architecture in terms of its adequacy for the requirements as well as in terms of the consistency of the implementation with the architecture.
ARCHITECTURE-CENTRIC ENGINEERING SOLUTIONS: IESE’S SERVICES

No two projects are alike. Our starting point is always your concrete context and your goals. We derive appropriate project goals and select the right services from our portfolio. Our services and our project approach are finally tailored to best fit your context and goals.

Architecture Requirements Elicitation
We elicit those requirements that are significant for your architecture and have a high impact on the architecture. This is important to have a sound basis for the definition and evaluation of your architecture.

Implementation Guidance
We help you to derive high-quality, compliant implementations of your architecture. To this end, a key factor is to provide your developers with pragmatic and usable documentation of the architecture.

Organizational Coordination and Planning
We help you to align your development process with your architecture and show you how system development should be organized in terms of time, organizational units, and developers.

Compliance Checking
With the help of Fraunhofer SAVE, we can check how compliant your architecture is with the actual implementation. High compliance is a key factor for sustainable evolution and maintenance.

Architecture Recovery
We help you to recover your architecture from the implementation if documentation is missing or if the implementation has significantly diverged over time. With the help of Fraunhofer SAVE, we can recover your architecture systematically and according to clear targets.

Architecture Capability Improvement
We help you to improve your architecture capabilities and practices by offering tutorials, coaching, and long-term transfer activities.

Architecture Construction
We support you in defining the architecture for your next generation products or major evolutions. With our methods for architecture design, modeling, and documentation, we collaborate with you on defining the architecture that best fits your requirements.

Architecture Prototyping
We increase your confidence in architecture decisions and demonstrate the feasibility of architecture ideas via early prototyping of your architecture. The result can serve as a sample for the realization of the system.

Technology Selection Support
We support you in making the right decisions regarding technology. For this purpose, we neutrally analyze architectural aspects and implications, which then serve as clear facts for technology selection, usage, or retirement.

Solution Adequacy Assessment
We assess for you how well your architecture can fulfill your key functional requirements and quality requirements. This can be done very early in the development lifecycle to reduce risks such as project delays or inappropriate systems. Further, it can strongly support migration and evolution decisions.
In the context of architecture-centric engineering, there are many situations that can only be mastered with the support of appropriate software tools. This is why our architecture-centric approaches come with a tool set that enables architects to deliver results in their day-to-day work.

We use Enterprise Architect by Sparx Systems as a central tool for creating and maintaining architectural models. We either use the built-in functionalities or provide own extensions and tailoring. By using Enterprise Architect, we enable customers to create models that can be utilized for particular purposes, such as impact analyses or customized documentation generation. Of course, our methods and techniques can also be adapted to other tools as well, such as Rational Software Architect, Magic Draw, or TOPCASED/Papyrus.

In order to keep the architectural model synchronized with system implementations, we have developed the Fraunhofer SAVE tool suite (the acronym stands for Software Architecture Visualization and Evaluation). SAVE is an award-winning state-of-the-art tool for analyzing and visualizing software architectures.

Its primary feature is architecture compliance checking, i.e., assessing the degree to which an implementation conforms to its intended architecture. SAVE - developed as an Eclipse plug-in – reverse-engineers system artifacts (e.g., source code written in Java, C/C++, C#, Delphi, Fortran, Ruby on Rails, etc.) and lifts this information to the abstraction level of the architecture (either automatically or using expert input). SAVE realizes a set of compliance checking algorithms for structural or behavioral views and among variants. Its powerful visualization offers many abstraction, navigation, and filtering capabilities, which accelerate the analysis of large-scale software systems.
IESE’s core competencies and services in software architecture are widely applicable to any domain in software systems development. In order to optimally address our customers’ challenges, we specialize our competencies and services for Embedded Systems and Information Systems, which means understanding the typical challenges, solution concepts, technologies, and standards of these two types of systems.

In Embedded Systems, we work in domains such as automotive and transportation, medical systems, and automation and plant engineering. From a product perspective, key challenges of embedded systems are resource consumption, real-time performance behavior, and safety aspects. In terms of processes, organizations developing embedded systems usually have to deal with compliance with particular standards, or challenges with respect to the integration of functionalities developed by different sub-contractors. With our specialized quality models, solution concepts, and expertise in standards like AUTOSAR or ISO 26262, we support you in achieving standard compliance and in tackling your particular challenges related to embedded systems development.

In Information Systems, we work in areas such as finance and insurance, airlines and logistics, eGovernment, and agriculture. Key challenges for large-scale, long-living information systems are flexibility, interoperability, and user experience. We explicitly cover such challenges with quality models and solution concepts. In information systems, there are widely-known paradigms like service-oriented architecture (SOA), business rule management (BRM), or cloud computing, which are supported by multiple technologies. We help you to understand the essence of these paradigms from a business, architecture, and technology perspective.

The vertical integration of Embedded Systems and Information Systems leads to a new area where architectures play a central role. Our research activities address how business processes can consume real-time sensor data, how networks of several embedded systems can interoperate in cyber-physical systems, and how modern metaphors like serious gaming help non-engineers in running and controlling the complex symbiosis of merged embedded and information systems.
“In more than 30 cases, an architecture evaluation of IESE allowed identifying and rating risks early. Based on the results, business-critical decisions were facilitated.”

“The architecture evaluation of IESE offers great scalability in terms of effort and time spent. First results can be achieved very quickly. Depending on the desired breadth and confidence of the results, IESE conducted projects from 5 person-days up to 200 person-days.”

“More than 90% of the recommendations resulting from an architecture evaluation have been accepted by a customer and their realization was ordered within 2 weeks”

“In more than 80% of the cases, no up-to-date or sufficient architecture documentation exists, which is a basis for making critical decisions. The IESE architecture reconstruction supports goal-oriented and efficient reconstruction in order to achieve a sound basis for decision making.”

“With Fraunhofer SAVE, many systems with several million lines of code were checked for compliance to the intended architecture. In particular systems, more than 50,000 violations of the architecture were discovered and recommendations for improvement were derived.”

“In many cases, the IESE architecture evaluation helped to make fact-based and sustainable technology selections.”

“Optimization of the visualization of software architectures in Fraunhofer SAVE allowed reducing the processing time of analysis tasks by more than 60%.”

“A recurring check of the architecture compliance of a product family’s implementation with Fraunhofer SAVE allowed increasing conformance sustainably up to 98%.”

“Architecture-centric development reduced the effort for implementation and testing by 35% in a customer company, preserving at least the same quality.”

“In architecture coaching with customers we could observe that during architecture construction, far more than 50% of the architects’ time had to be spent on requirements clarification, re-construction of existing system ideas, and for other communication and organization tasks.”

“With goal-oriented analysis and consequent re-design of the architecture, the response time of user interactions in an architecture visualization tool was reduced by a factor of more than 10 for very large systems being visualized.”

“Developers getting real-time feedback about architecture violations by SAVE LiFE cause up to 60% less architecture compliance violations.”
**Wikon GmbH** is a manufacturer of remote control systems (hardware and software). One of their main applications are battery-powered remote control systems for the energy sector. We iteratively introduced architecture-centric development at Wikon GmbH. The results show that investments into reuse and architecting are worthwhile, even for small development organizations. The shift of the development paradigm from ad-hoc reuse towards pro-active, systematic reuse can be considered a success. For the first product generation developed, architecture-centric development saved 12 person-months of development time and 3 person-months (both approx. 35%) for quality assurance (compared to the previous product generation). At the same time, no decrease in quality was observed for the first products in the field.

**Tekla** is a Finnish company developing software products for the engineering and construction market. Tekla Structures is one of Tekla’s key products. In order to maintain its prominent market position, Tekla is continually improving their software in terms of new features and invests into high quality. As part of this strategy, Tekla performed an architecture evaluation together with Fraunhofer IESE. The architecture evaluation results allow Tekla to make improvement efforts in a very dedicated and focused manner. Tekla also develops new products aimed at other user groups, and in doing so, follows an agile software development process. Fraunhofer IESE supported Tekla in aligning its architecture practices and its agile development process.

For this purpose, IESE’s architecture method ACES was tailored to Tekla’s context and introduced via a combination of tutorial and coaching. Tekla’s architects successfully adopted this approach, which resulted in more predictable product quality.

**John Deere** is one of the leading manufacturers in the world regarding agricultural, forestry, and construction machinery. Together with John Deere engineers we analyzed the software architectures of modern display systems, making it possible to derive improvement measures based on which both the complexity and the diversity of product variants can be managed. John Deere was able to incrementally address the derived improvement measures with the support of Fraunhofer tools.

“Collaboration with Fraunhofer IESE experts was really useful for us. With theoretical and practical examples they helped us to understand how we can improve our way of doing architecture work and to build capabilities for the future. Tekla is happy with the co-operation and will continue working with Fraunhofer IESE in the future.”

Timo Rihntiemi, Manager Product Architecture at Tekla

“Collaboration with Fraunhofer IESE was involved in the conceptual phase of system development. Future-oriented technical realization possibilities were analyzed and a neutral assessment of the existing product was performed in accordance with our requirements. The performance was impressive, since a solid basis for how to proceed further could be created on a very professional level within the shortest possible time. Our thanks go to Fraunhofer IESE for this successful collaboration!”

Michael Nachtigäller, Deutsche Lufthansa AG Robert Bade, Lufthansa Cargo AG

--

**Architecture Centric Engineering Project References**

“**The future-oriented applications in agricultural technology require highly interconnected, reliable, and flexible software systems. Fraunhofer IESE is our preferred partner when it comes to structuring and introducing new software architectures and system concepts.”**

Dr. Thomas Engel, Director of John Deere ISG Europe

“In the context of the selection process for replacing an operative system, Fraunhofer IESE was involved in the conceptual phase of system development. Future-oriented technical realization possibilities were analyzed and a neutral assessment of the existing product was performed in accordance with our requirements. The performance was impressive, since a solid basis for how to proceed further could be created on a very professional level within the shortest possible time. Our thanks go to Fraunhofer IESE for this successful collaboration!”

Michael Nachtigäller, Deutsche Lufthansa AG Robert Bade, Lufthansa Cargo AG
Software is a part of our lives. Embedded into everyday equipment, into living and working environments or modern means of transportation, countless processors and controllers make our lives simpler, safer, and more pleasant. We help organizations to develop software systems that are reliable in every aspect, and provide empirical proof of the necessary processes, methods, and techniques, emphasizing engineering-style principles such as measurability and transparency.

Fraunhofer IESE is one of the worldwide leading research institutes in the area of software and systems development. A major portion of the products offered by our collaboration partners is defined by software. These products range from automotive and transportation systems via automation and plant engineering, information systems, health care and medical systems to software systems for the public sector. Our solutions allow flexible scaling. This makes us a competent technology partner for organizations of any size – from small companies to major corporations.

Under the leadership of Prof. Dr. Dieter Rombach and Prof. Dr.-Ing. Peter Liggesmeyer, the past decade has seen us making major contributions to strengthening the emerging IT location Kaiserslautern. In the Fraunhofer Information and Communication Technology Group, we are cooperating with other Fraunhofer institutes on developing trend-setting key technologies for the future.

Fraunhofer IESE is one of 60 institutes of the Fraunhofer-Gesellschaft. Together we have a major impact on shaping applied research in Europe and contribute to Germany’s competitiveness in international markets.
CUSTOMERS OF FRAUNHOFER IESE

Adam Opel GmbH
Airbus Deutschland
Alcatel SEL AG
Audi Electronics
BMW Group Forschung und Technik GmbH
BASF SE
BearingPoint GmbH
B.Braun
CAS Software AG
Continental Teves
Deutsche Bahn AG
Deutsche Lufthansa AG
EADS Deutschland GmbH
Fiducia IT AG
Finanzinformatik
Ford Forschungszentrum
Fujitsu
HegerFerrit GmbH

Hella KGaA Hueck & Co
Hyperware
IDS Scheer
IHK Darmstadt Service GmbH
Infinion Technologies
INSPIRE AG
John Deere
KEIPER GmbH &Co. KG
KOBL Systems GmbH
KBS AG
Lufthansa Systems
Microsoft Deutschland
Mitsubishi Research Institute
Motorola
Nokia Networks
OrgaTech Unternehmensberatung
Philips Electronics
Porsche engineering Group
Porsche Informatik

Ricoh
Robert-Bosch GmbH
Roche Diagnostics
Rolls-Royce
SAP AG
Siemens AG
Software AG
Sopera GmbH
Süddeutsche Klassenlotterie
Terex Demag GmbH
Testo AG
T-Mobile International
TÜV-SÜD
Tekla Oy
TXT e-solutions
Wikon GmbH
Volkswagen AG
XING AG

31
The division Information Systems Development develops innovative methods and solutions for the development of complex information systems and interactive systems. In order to offer an immediate added value, we apply our methods directly during our customers’ product development and/or transfer our methods and solutions to our customers. These include organizations that use software solutions for their business processes, e.g., from the banking and insurance sector or from the domain of medical IT systems, organizations from such areas as public administration and defense, as well as software developing companies (both large corporations and SMEs).

We address challenges arising during development early and continuously with the help of constructive, innovative development approaches and methods, from requirements elicitation via architecture to design. Our strength lies in examining the non-functional (quality) requirements, which are hard to capture and often affect the entire system. Clearly defined software development artifacts with clearly defined levels of abstraction allow focusing on what is really important and interlink with both traditional and agile process models.

Requirements Engineering
We support you with our renowned requirements approach Satisfy, which is tailored to your needs and goals and ensures traceability, integrated usability, and minimal documentation of the necessary requirements decisions.

User Experience for Business Applications
With our proven methods we support you in developing systems that not only support their users, but also allow having a positive User Experience. This has been proven to contribute to the achievement of your business goals.

Architecture-Centric Engineering
With our successful Fraunhofer ACES approach, we support you in defining and evaluating architectures for complex information systems. The iterative, scenario-based process allows you to master the complexity of architecture definition and migration and to make reliable predictions regarding critical features of the system early on.

Variation Management
We support you in establishing, developing, and managing software product lines as well as in analyzing and improving your already existing configuration infrastructure for variation-rich information systems. Our services build upon our successful product line engineering approach PuLSE™ and extends it to support several other variation management approaches.

Business goes Mobile
With our product GoMobile we help you to exactly identify those areas where a mobile app makes sense as support for your business processes. We can also develop a prototype that can be executed on various mobile devices to help you make your decisions.
Hidden in transportation systems, medical devices, consumer goods, and almost all other technical products, embedded systems are performing essential tasks that make our daily lives safer and more comfortable. Every year, more than three billion embedded components and devices are manufactured, incorporating 98% percent of all microprocessors built. Embedded systems are omnipresent, and our modern economy and society would be unable to survive without them.

In order to be able to develop such highly reliable and safe systems in a cost-efficient manner, development must focus on system quality right from the outset. This means that efficient support is needed for the developers to allow them to keep track of everything, despite rapidly increasing system complexity, and to make the right decisions based upon facts. The ESD department supports its customers during the entire development cycle from requirements to implementation.

**Model-based Development**
If we want to model not only software, but entire systems including the relevant non-functional properties, “off-the-shelf” technologies often do not suffice. When it comes to selecting, adapting, and combining suitable languages and tools, the experts of Fraunhofer IESE will provide comprehensive support for model-based development.

**Architecture-Centric Engineering**
The system architecture influences quality and development costs across system generations. Using our Fraunhofer ACES approach, systematic processes in combination with automated, tool-supported architecture analyses and simulations make it possible to assess existing architectures and to identify and assess possible architecture variants early on. Architecture faults can thus be avoided proactively already during development.

**Variation Management**
Embedded systems are often not single products, but rather product families. They must be extensible and changeable throughout their lifetime. The strategic planning of product variants and modifications enables reuse rates of over 90%. Thus, system quality increases while costs and development times decrease. The ESD department supports its customers with efficient methods for variation management – from the initial potentials analysis to the introduction of customized solutions in the organization.
Embedded Systems Architecture
Dr. Thorsten Keuler
Tel.: +49 631 6800 2162
thorsten.keuler@iese.fraunhofer.de

Vertical Integration Architecture
Dr. Jens Knodel
Tel.: +49 631 6800 2168
jens.knodel@iese.fraunhofer.de

Information Systems Architecture
Dr. Matthias Naab
Tel.: +49 631 6800 2249
matthias.naab@iese.fraunhofer.de

Fraunhofer Institute for
Experimental Software Engineering IESE
Fraunhofer-Platz 1
67663 Kaiserslautern
Germany