Trends in Embedded Software Development in Europe

Dr. Dirk Muthig

dirk.muthig@iese.fraunhofer.de

Problems

A software project exceeds the budget by 90% and the project time by 120% in average

- Project Management
  - Incapable or even no project management
- Development processes
  - Bad planning and estimation of project activities
- Engineering practices
  - Incomplete product specifications Interdisciplinary nature is not considered sufficiently
  - Lack of sound architectures
  - Lack of standardized procedures and mechanisms

Fraunhofer Institut Experimentelles Software Engineering

Outline – Trends

- System Engineering (Systems of Systems)
- Software and System Architectures
  - Definition and Prototyping
  - Effects on Organization and Projects
- Distributed Development: Smart Globalization
- Model-driven Development
- Open Source Software
- Quality Engineering: Efficient Construction of Quality
- Product Line Engineering
  - Efficient production of system families through reuse

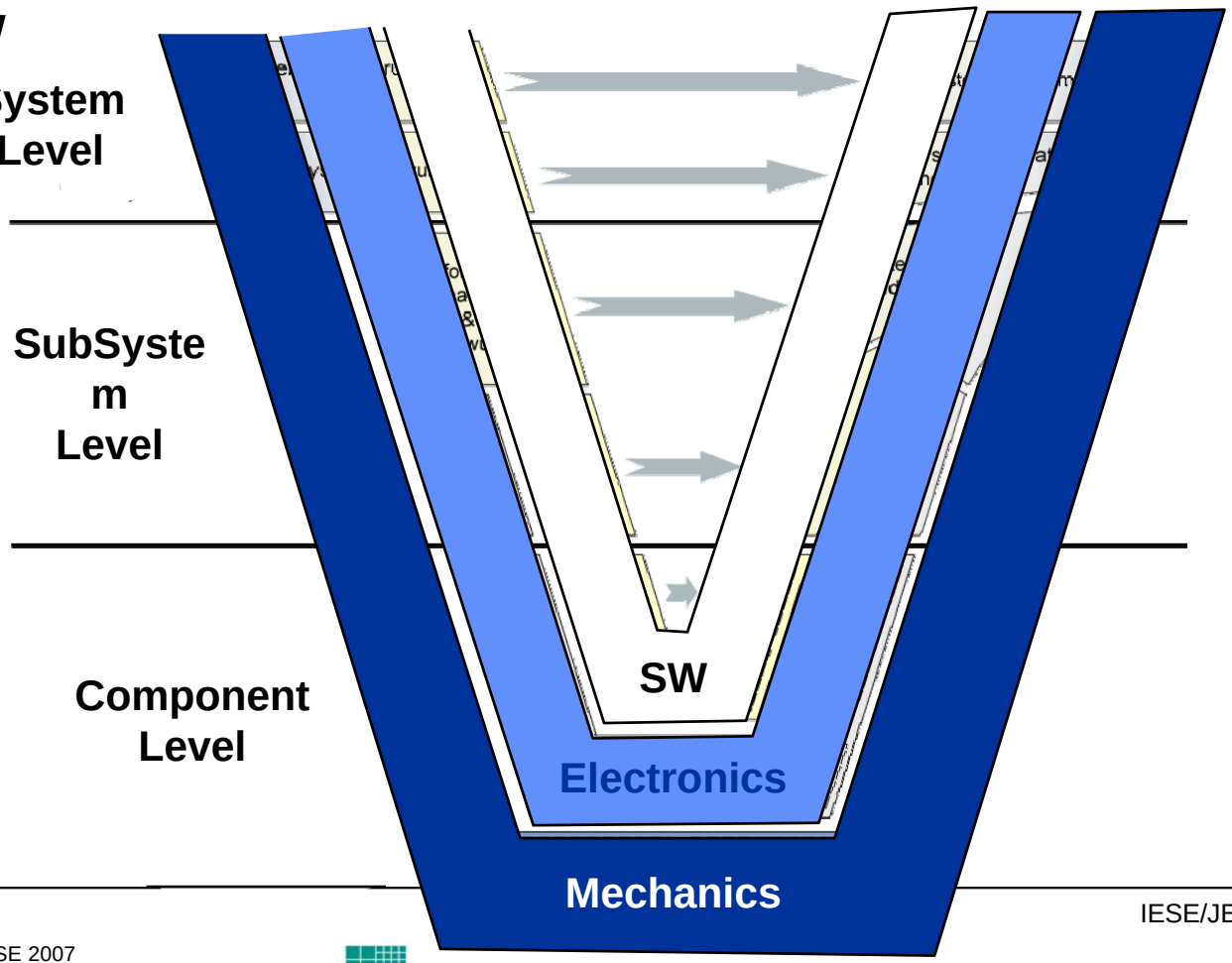
Fraunhofer Institut Experimentelles Software Engineering

System Engineering – Process Model [EQUAL]

- Process model for system engineering
  - Aligned with the V-Model (left-hand side: constructive, right-hand side: analytical)
  - 3 levels of abstraction describe the solution space
  - Subdivision of different levels into subsystems under consideration of common strategies like minimization of coupling
- Experience
  - Better communication and thus better alignment in between developers
  - Early detection of mismatches
  - Higher transparency and goal-orientation in the project

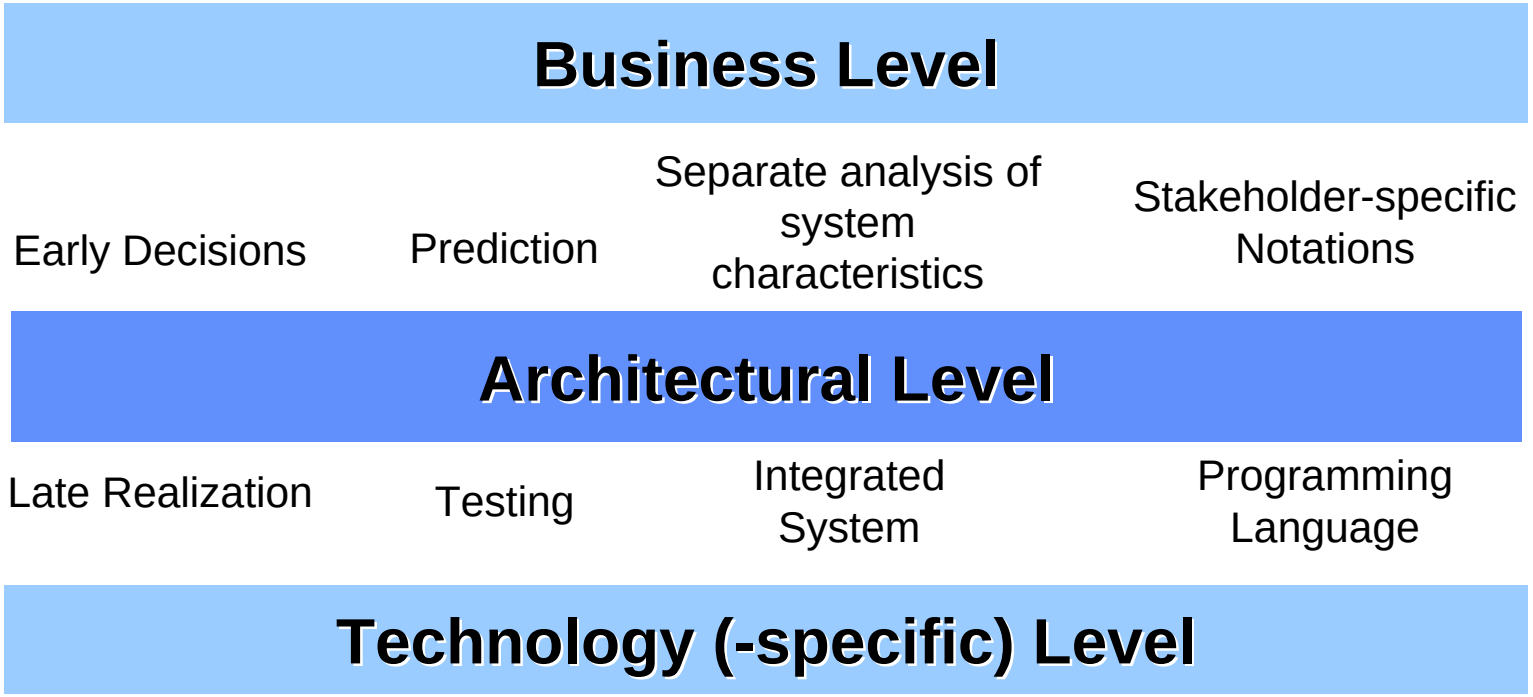
Fraunhofer Institut Experimentelles Software Engineering

Overview



Fraunhofer Institut Experimentelles Software Engineering

Architecture as Bridge between Problem and Solution Space



Fraunhofer Institut Experimentelles Software Engineering

Software Architecture – State of the Practice

- All software systems developed so far have a software architecture, but
  - it typically has not been consciously designed, and
  - there often exists no explicit documentation describing it
- Architecture descriptions are typically
  - ambiguous
  - hard to understand
  - inadequate
  - incomplete
  - not up-to-date
- Most common diagram: layering structure
- Architecture not used as the basis for other activities such as quality assurance, reengineering, configuration management

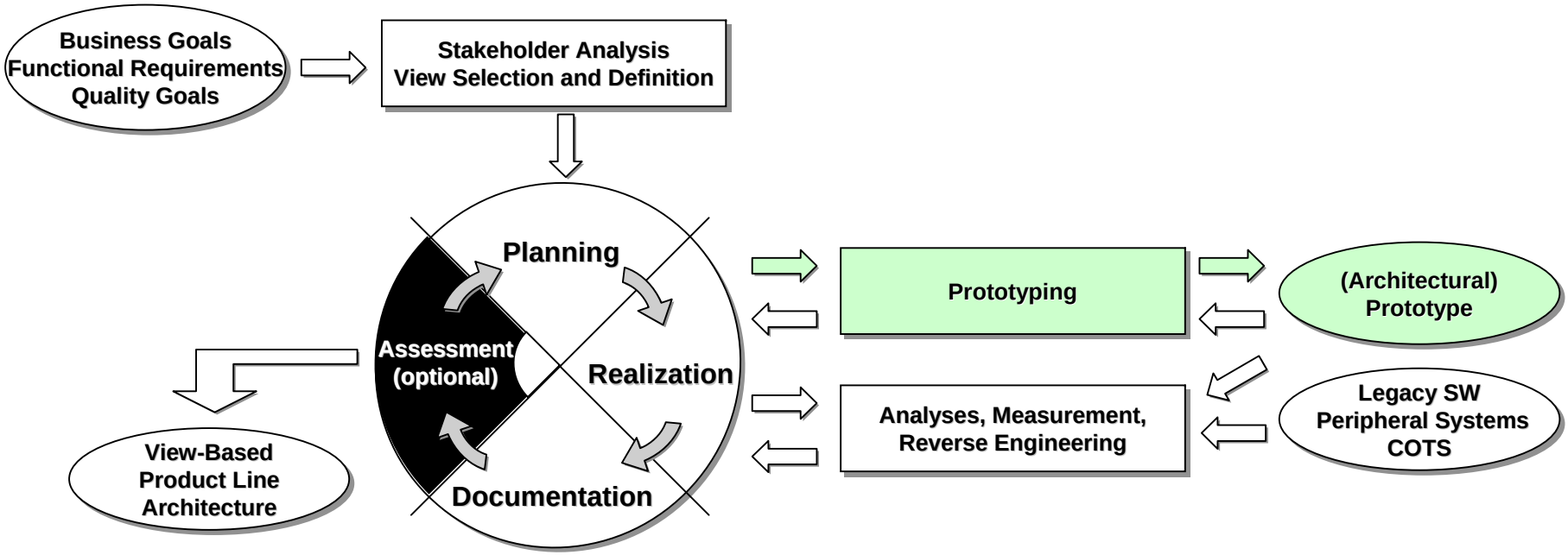
Fraunhofer Institut Experimentelles Software Engineering

Architecture – Central Role to Projects

- Link between problem and solution space
  - First design artifact addressing quality requirements such as maintainability, performance, reliability, etc.
  - Provides a means to express, negotiate, and resolve competing concerns of the various stakeholders of a system
  - Serves both technical and organizational purposes
- Basis for a number of activities such as
  - Project planning and management,
  - Component engineering
  - Quality engineering,
  - Configuration management

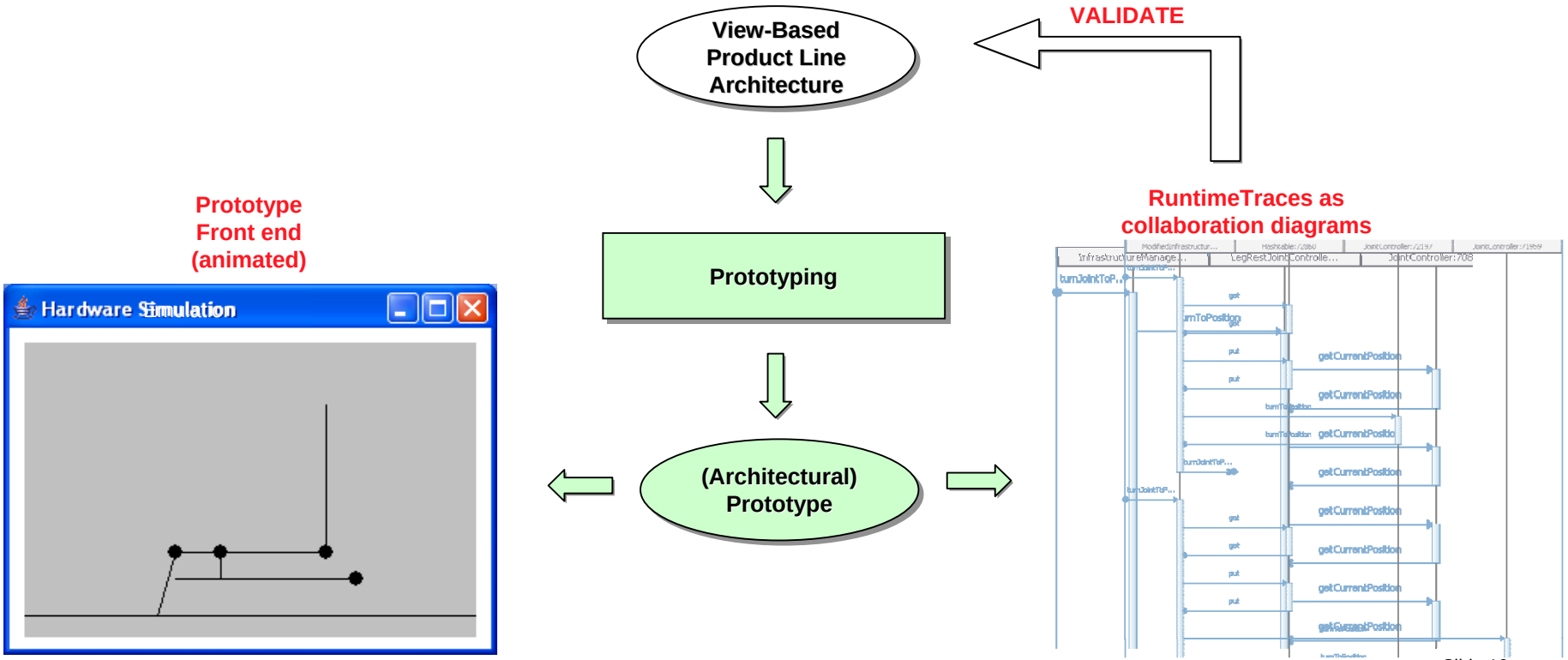
Fraunhofer Institut Experimentelles Software Engineering

Architecture Definition Process [Fraunhofer PuLSE™]



Fraunhofer Institut Experimentelles Software Engineering

Prototyping – Validation of Architectural Collaborations (Car Seat)



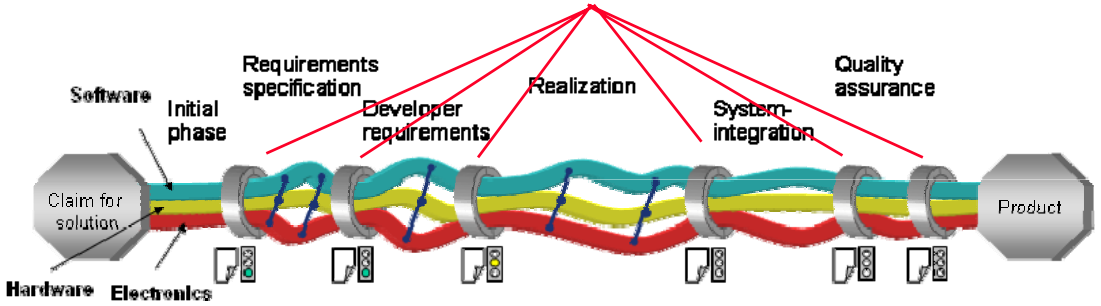
Effect of Architecture on Structure of Development Organization – and vice versa

- Architecture prescribes software units that must be integrated
- Units of software correspond to work assignments
- Architectures dictate an organizational structure for development and maintenance of the system
- Hence, architecture definition must consider organizational structures
- Today, distributed development is the normal situation (rather than an exception)
  - Use resources in the best possible way with respect to their capabilities and expertise
  - Carefully assign components centered around mission- and organizational-critical aspects

Fraunhofer Institut Experimentelles Software Engineering

Project Planning and Management

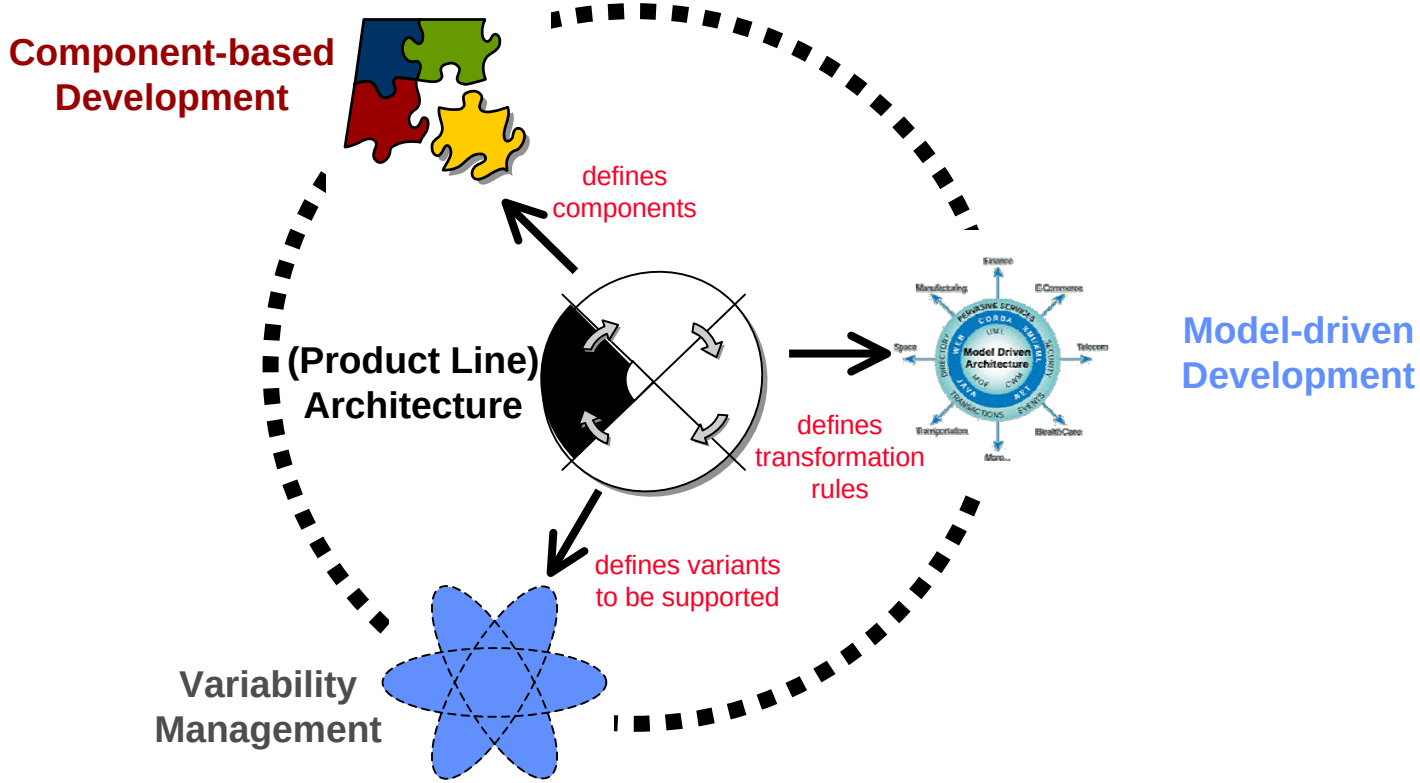
- System architecture provides …
  - Common understanding of product (i.e., system)
  - Consistent and interdisciplinary product specification
  - Integrated view still using interdisciplinary description techniques (i.e., stakeholder-specific notations)
- (Software) architecture defines …
  - Work assignments to be carried out by separate design and implementation teams
  - Hence basic structure of project activities
  - Useful synchronization points among results of teams that are working in parallel (Quality Gate Concept)



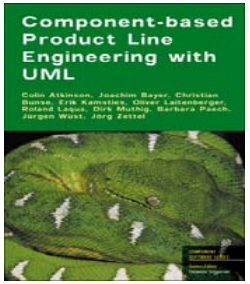
[ProMiS]

Fraunhofer Institut Experimentelles Software Engineering

Vision – Model-driven Development

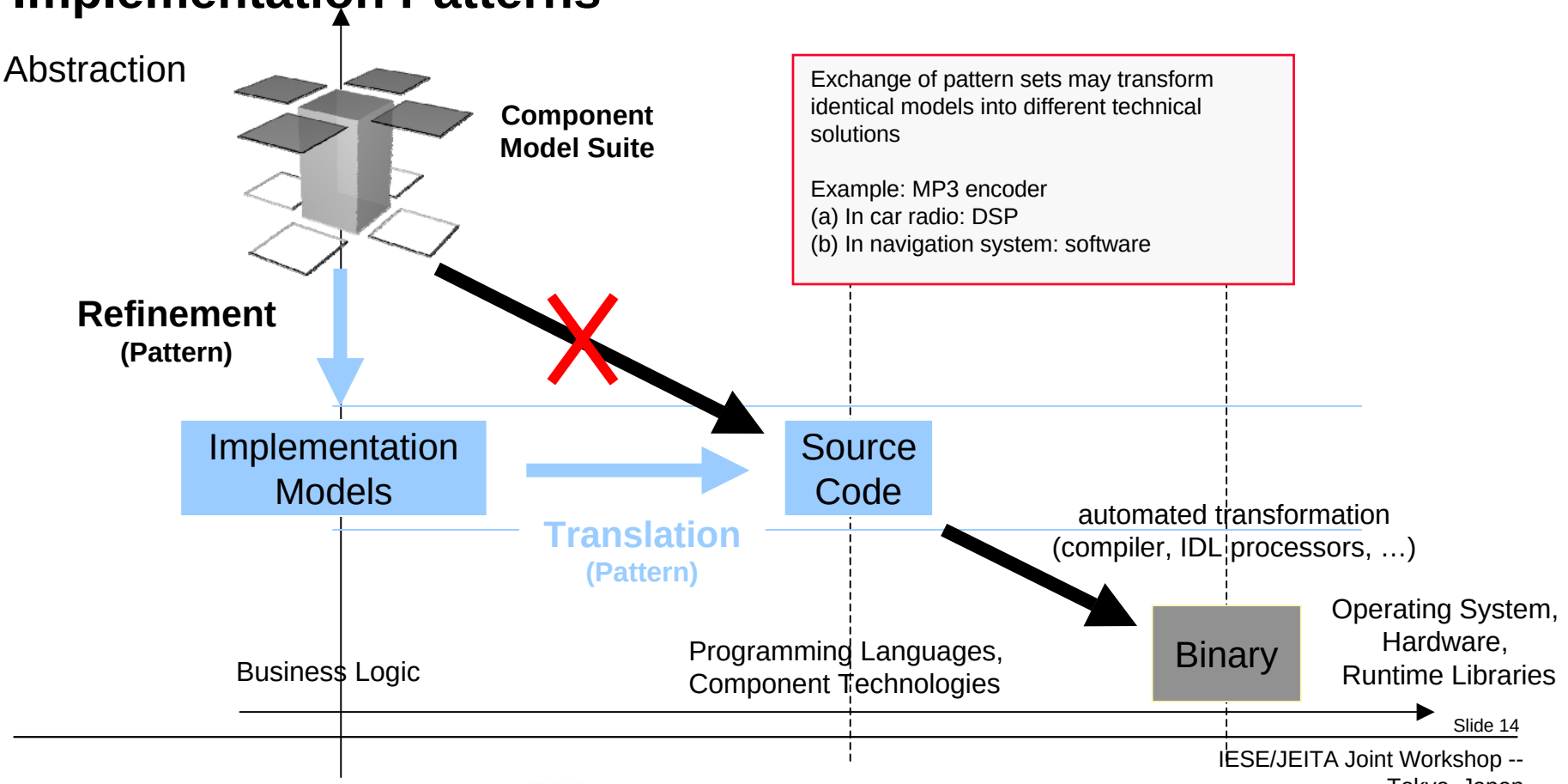


[KobrA method]



Fraunhofer Institut Experimentelles Software Engineering

Implementation Patterns



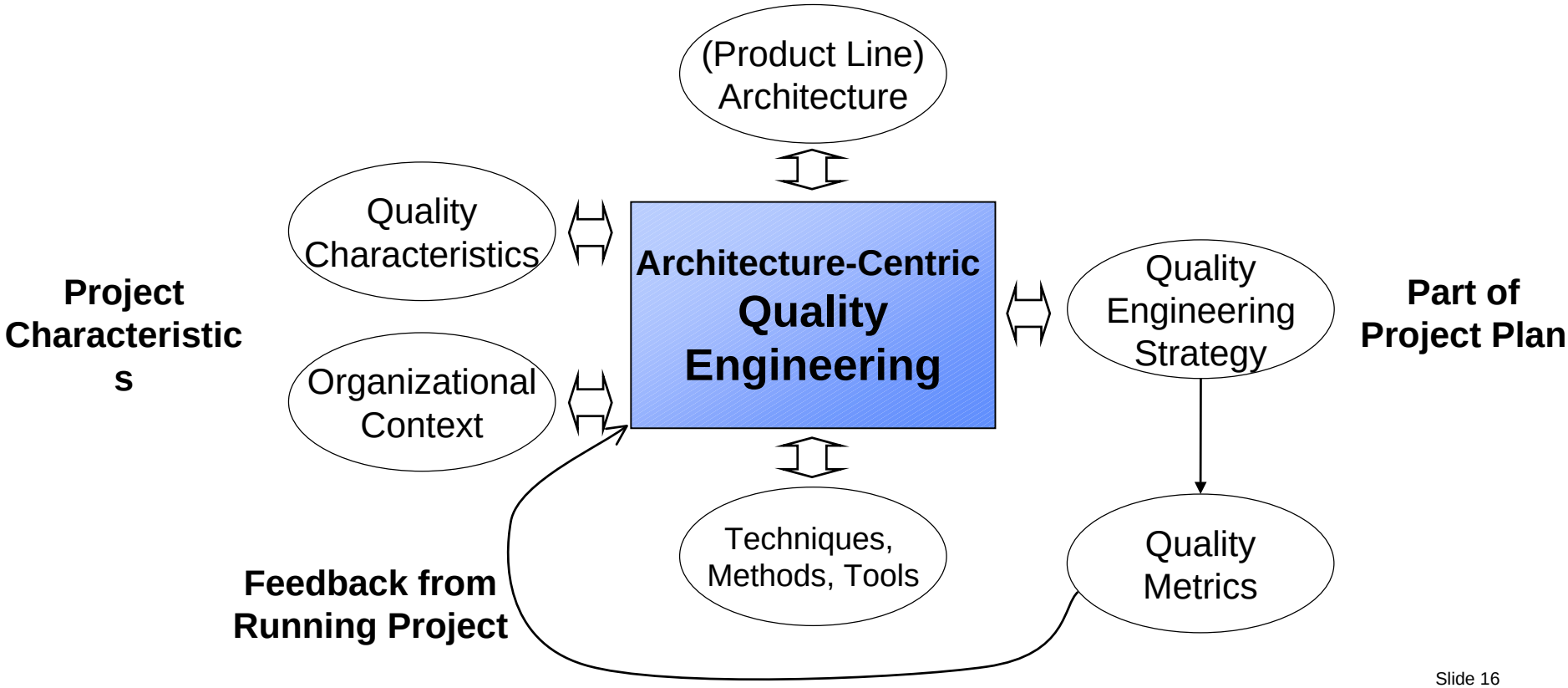
Fraunhofer Institut Experimentelles Software Engineering

Quality Engineering – Industrial Requirements

- Fully compatible to existing environments
  - By defining quality profiles wrt. individual systems, contexts and organizations
- Defines customized (quality) engineering strategies matching needs of particular projects
- Guaranties satisfaction of individual quality profiles
- Optimizes for
  - Quality profiles
  - Consumption of effort and time

Fraunhofer Institut Experimentelles Software Engineering

Quality Engineering [Fraunhofer ArcQue™]



Fraunhofer Institut Experimentelles Software Engineering

Architecture-Centric Quality Engineering

- Recursive Unit Testing
  - Component tests => validated building block
  - Integration test => validated component
    (Note: one man‘s system is another one‘s component)
- Balance QE effort among
  - Organizational units
  - Similar Projects
  - “for-reuse” and reuse activities
- Note: Reusing component quality requires stable context, that is, stable architecture!

Product Line Software and System Engineering

Product line engineering stands for pro-active, strategic, and successful reuse

- Challenges in industry
  - Need for reducing cost, effort, and time-to-market
  - Complexity and software size increases
  - Increasing request for
    - Quality software and
    - Individually customized products
- Envisioned Solution: Large-scale Reuse
  - Reusing existing software saves time and effort
  - Reusing approved software brings quality
  - Focusing on new functionality enables realization of individual requirements

Fraunhofer Institut Experimentelles Software Engineering

Summary

- Dominant and critical role of software in the context of embedded systems is more and more accepted in industry
- Professional ways for producing software-intensive systems are required
  - Predictable quality: software engineering
  - Efficiency: **product line engineering**
- Careful analysis of ROI and expected benefits, as well as measurement-based improvement strategy, are recommended
  - Prototyping: methods or products
  - Pilot projects

Fraunhofer Institut Experimentelles Software Engineering

Thank you for your attention!

Dr. Dirk Muthig
Division Manager „Software Development“

Fraunhofer IESE
Fraunhofer-Platz 1
D-67663 Kaiserslautern, Germany

Tel: +49 (631) 6800-1302
Email: dirk.muthig@iese.fraunhofer.de

Fraunhofer Institut Experimentelles Software Engineering

Open Source Software

- Companies have investigated diverse business models based on open source software
  - Tools
  - Components
  - Products
  - …
- No common, proven model established yet
- Note that concepts and processes associated with open source software are also naturally relevant to any (distributed) organization

Fraunhofer Institut Experimentelles Software Engineering

Open Source Tools – Example

- Software in Avionics:
  - A330/A340 12Million LOC, not much improvement in productivity since 1980s (introduction of code generation)
- Focus: tool market for dependable embedded systems
  - Long-term availability: e.g., A300 started in 1972, support until 2050!
  - Big issue: Volatility of tool market: vendors change every 2 or 3 years
    (e.g. AUTAN> ATTOL> RTRT) or disappear (Object/GEODE)!
  - Too many tools in stage of prototypes or not widely used (small market)
  - Tools create high deployment costs for distributed engineering
- => open source tools, Open Source strategy for avionics software development (e.g., www.topcased.org)

Fraunhofer Institut Experimentelles Software Engineering