Corporate Technology

What an Architect Needs to Know

Experiences from the Siemens Curriculum for Software Engineers

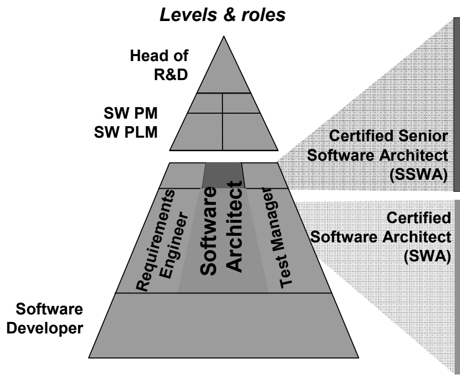
Frank Buschmann
Siemens AG
Corporate Technology
Systems Architecture and Platforms

Content

- **Motivation**
- **Software architect target profile**
- **Senior software architect curriculum**
- **Summary**

Background and motivation

The economic success of many Siemens products is highly dependent on key software knowledge and practices. The qualification of software engineers has high priority for Siemens



Class A Project:

- High complexity (platform / product line)
- High degree of innovation
- Big business impact, high risk
- Cross-functional, distributed structure, big team

Class B Project:

- Moderate innovation in technology, medium risk
- Medium business impact, medium teams

Class C Project:

- Enhancing known technology and requirements, low risk
- Single site development, small teams

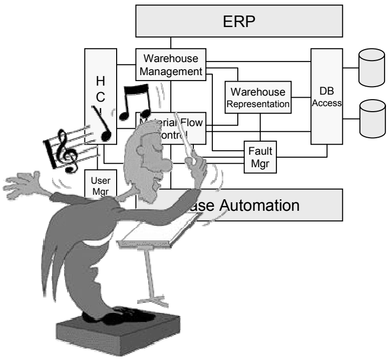
Content

- **Motivation**
- **Software architect target profile**
- **Senior software architect curriculum**
- **Summary**

Software architect mission

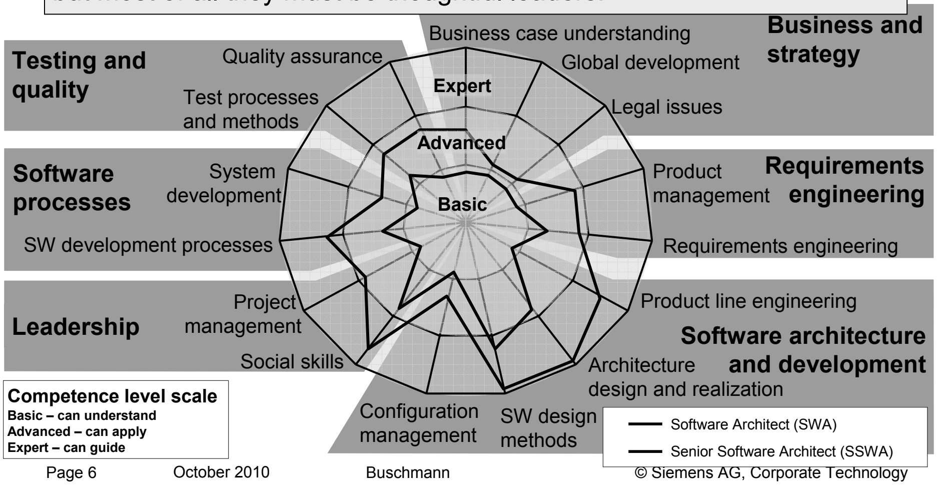
Software Architects drive and guide the specification and realization of a software system through its entire lifetime

- They **provide the architecture vision** of a product
- Their **decisions are driven by clear focus** on the intended business for the software and its associated requirements
- They are **guided by vision and experience**
- They **lead, guide, coach, and motivate** the architects and developers in their teams
- They **involve the relevant management** before taking important decisions



Software architect competence spider

Architects must be proficient in all areas of software development, but most of all they must be thoughtful leaders!



Involvement in the business case

A properly defined business case and project scope inform about

- The key purpose and main responsibilities of a software system
- Its business-relevant requirements and USP’s
- The context of the system: boundaries, users and their interests, and the domain model
- The system’s business model
- The envisioned market for the system and the expected market share
- The expected revenue (over time), the planned investment (over time)
- The system’s technology, development and release roadmaps
- The system’s Intellectual Property Rights (IPR) strategy

The architect must understand the system’s business case, and is involved in defining its scope, domain model, roadmaps, and IPR strategy

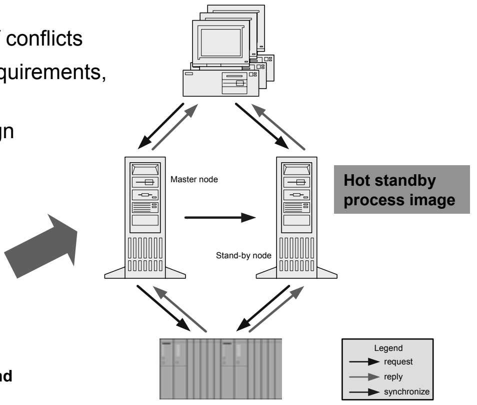
Involvement in requirements engineering

An architect is dependent on clearly defined requirements to take explicit, thoughtful, focused, and balanced design decisions.

- Requirements drive the selection of appropriate design concepts and realization technologies for their solution
- They guide design decisions in case of conflicts
- They ensure a design addresses all requirements, and only the requirements
- They enable testing the specified design and its realization

- **The planned uptime of the system is 7x24, the planned downtime is 10 consecutive days every 2 years**
- **The process image of the plant must be available 99,999% of the planned uptime**
- **Recovery [reference to the recovery process] in case of an unplanned outage should take at most 2 seconds**

Qualitative requirements lead to appropriate and sustainable design decisions

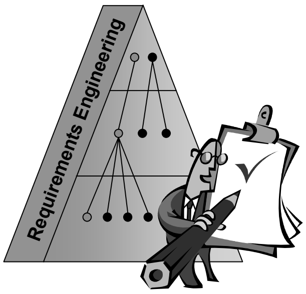


SIEMENS

Involvement in requirements engineering

To guide the design and implementation of a software system, requirements should expose the following quality attributes:

- Feasible – supports the business case
- Correct – precise qualitative description
- Unambiguous – precise qualitative description
- Testable – precise quantitative description
- Consistent – with other requirements
- Traceable – can be identified clearly
- Prioritized – regarding business value, technical risk, realization complexity, ...



The architect must understand the system's requirements, is involved in challenging their properties, and contributes all technical aspects to their specification and prioritization.

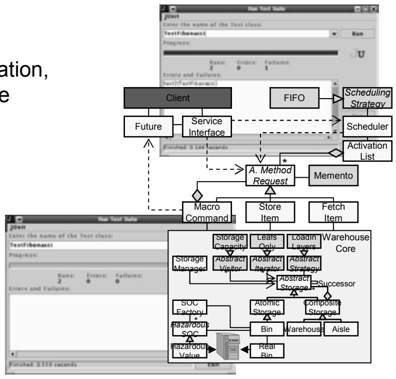
It is the architect's responsibility to initiate dialog with the relevant management and project roles if requirements are lacking the above qualities.

SIEMENS

Involvement in testing

Test can help tell an architect how sustainable an architecture is and how well the architecture meets the system requirements

- A defined **test strategy** supports testing the success critical aspects of the system
- **Design for testability** requires clear modularization, strict design by contract, and stable intermediate states along the control flow of key use cases
- Regular **architecture reviews** and **code quality management** help to maintain architecture sustainability and to avoid architecture drift

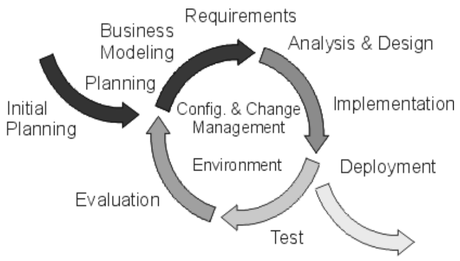


The architect and test manager must agree on a test strategy, the architect must prepare the system's architecture for testability, participate in reviews, and interpret and react on test results

Involvement in software processes

Define and realize a software architecture using an iterative, risk-driven, requirements-driven, and test-driven development process, in which

- An **iterative, time-boxed** approach provides continuous feedback
- **Risk- and requirements orientation** ensures that the most important aspects of the system's realization are addressed first: key functionality, quality, technological risks
- A **test-driven** approach provides concrete feedback on the quality of the architecture and its realization

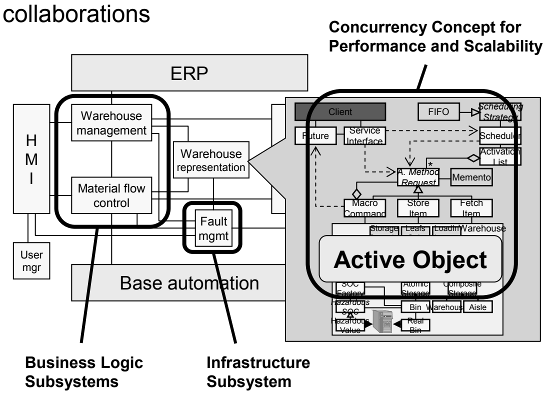


The architect is involved in defining the software development process to ensure that it defines a feedback loop for achieving product quality and less risk

Responsibility in design and realization

Define a sustainable baseline architecture – an architectural "whole":

- The fundamental structure and form of the software system: its core parts, their main responsibilities, relationships, interfaces, and collaborations
- The concepts for addressing success-critical, system-wide quality attributes
- The guiding principles and design directives for the architecture

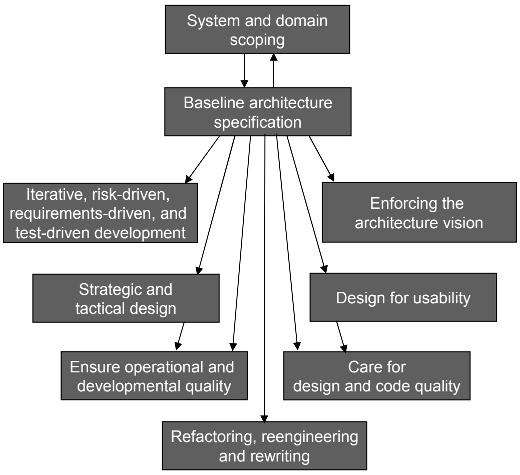


Key for success is that all architecture work is driven by requirements and that the architect guides realization: architect also implements

Responsibility in design and realization

An architect needs a clear set of values, activities, practices, and methods to

- Specify and implement a software architecture constructively and in a timely fashion
- Check and ensure the appropriate architectural quality
- Respond to changes of all kinds, such as changing requirements and priorities
- Deal with problems that arise during the definition and realization of the software architecture



System and domain scoping

Baseline architecture specification

Iterative, risk-driven, requirements-driven, and test-driven development

Enforcing the architecture vision

Strategic and tactical design

Design for usability

Ensure operational and developmental quality

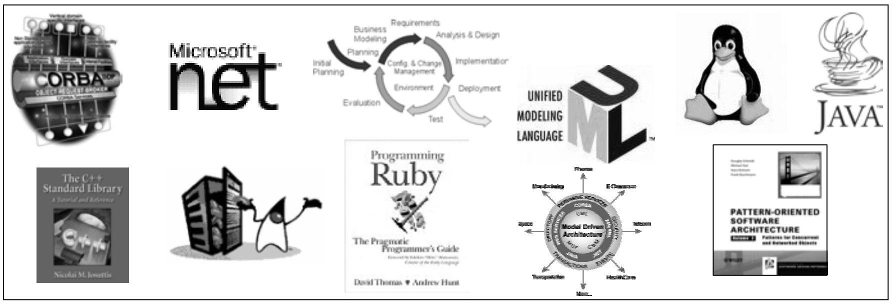
Care for design and code quality

Refactoring, reengineering and rewriting

Responsibility in design and realization

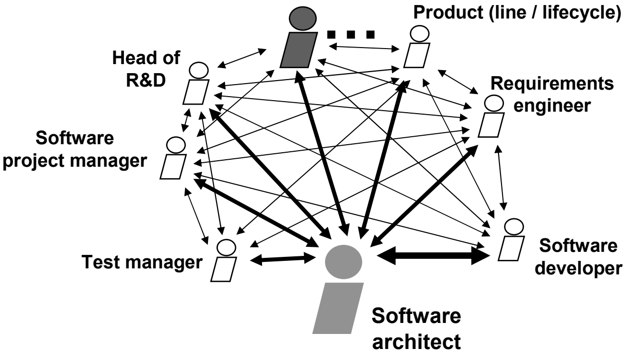
An architect needs a sound expertise in methodology and technology

- What are the methods and technologies all about and what is their range of applicability?
- What are their benefits and their costs?
- What methods and technologies are (competitive) alternatives and what methods and technologies are natural complements?
- How are the methods and technologies applied in detail?



General responsibility: leadership, collaboration, and interaction

Leadership is crucial: software architects interact with many other stakeholders in the software development lifecycle



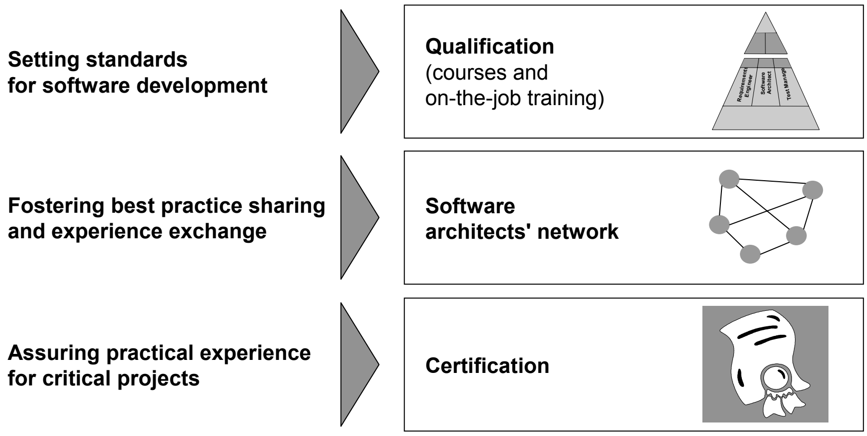
- Motivation
- Team building
- Decision enforcement
- Peer sparring
- Reviews
- Coaching
- Stakeholder-specific presentations

Content

- **Motivation**
- **Software architect target profile**
- **Senior software architect curriculum**
- **Summary**

Elements of the qualification program

A qualification program for software engineers requires more than a set of seminars and trainings



Setting standards for software development → **Qualification** (courses and on-the-job training)

Fostering best practice sharing and experience exchange → **Software architects' network**

Assuring practical experience for critical projects → **Certification**

Guiding principles (1)

The qualification program for senior software architects should address twelve guiding principles for development of software-intensive systems

1. Architecture is the key throughout the whole lifecycle and across releases
2. Build on existing basis where feasible (from technical and business perspective) and be able to recognize when such reuse is not suitable
3. Avoid unnecessary technological platform development by using technical standards and products available on the market
4. In product and system business the product (lifecycle) manager / in project and solution business the project manager is and must act as owner of the main requirement.
5. Pay particular attention to system quality attributes, often overlooked but are extremely important
6. Be prepared and able to handle changing requirements, but be aware about the risk of late changes

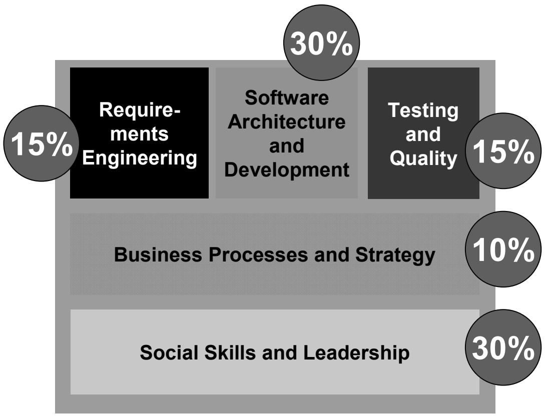
Guiding principles (2)

The qualification program for senior software architects should address twelve guiding principles for development of software-intensive systems

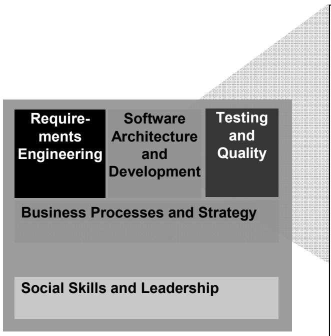
7. Synchronize well across the technical disciplines: software, mechanics, electronics, mechatronics, systems engineering
8. Work together truly as a team, avoid “silo” thinking, be willing and able to speak and understand the other roles and disciplines
9. Work iteratively (no 100% definition of all up-front), strive to identify and resolve technical and business risks early
10. Structure the system to avoid unnecessary complexity, and to actively enable and support multi-site development
11. Strive for transparency and base decisions on clear business / technical reasons, not political ones
12. Do not underestimate the importance of soft skills, these can be particularly important for convincing and motivating

Curriculum topics

The curriculum content partitions into five topic areas, each including a range of specific topics relevant for software architects

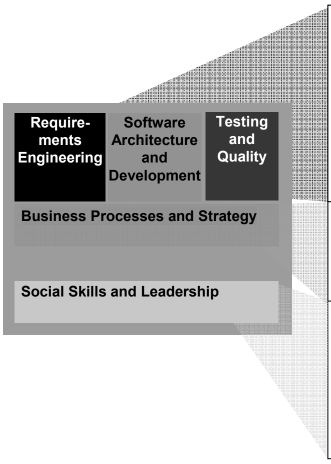

30%
Require-ments Engineering
15%
Software Architecture and Development
Testing and Quality
15%
Business Processes and Strategy
10%
Social Skills and Leadership
30%

Selected content details: architecture


Require-ments Engineering
Software Architecture and Development
Testing and Quality
Business Processes and Strategy
Social Skills and Leadership

- Domain modeling and system scoping
- Agile development practices
- Strategic and tactical design
  - Architecturally significant requirements
  - Walking skeletons
  - Variability management
- Design for operational and developmental quality
  - Patterns, practices, methods, technologies
- Design for usability
  - End user / customer acceptance
  - Developer habitability
- Enforcing the architecture vision (avoid architecture drift)
  - Communicating the architecture to stakeholders
  - Mentoring, coaching, reviews
- Refactoring, Reengineering, Rewriting

Selected content details: requirements, business, social skills

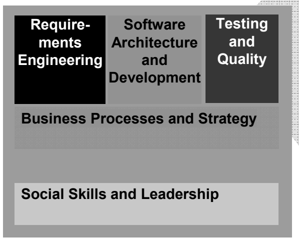

Require-ments Engineering
Software Architecture and Development
Testing and Quality
Business Processes and Strategy
Social Skills and Leadership

- Requirements management:
  - Elicitation
  - Documentation
  - Evolution
- Non-functional requirements
- Variability management
  - C/V Analysis

- Business case development
- Product definition
- Portfolio definition

- Intercultural aspects
- Conflict management; coaching
- Situation-dependent leadership
- Decision enforcement (in critical situations)
- Team Building

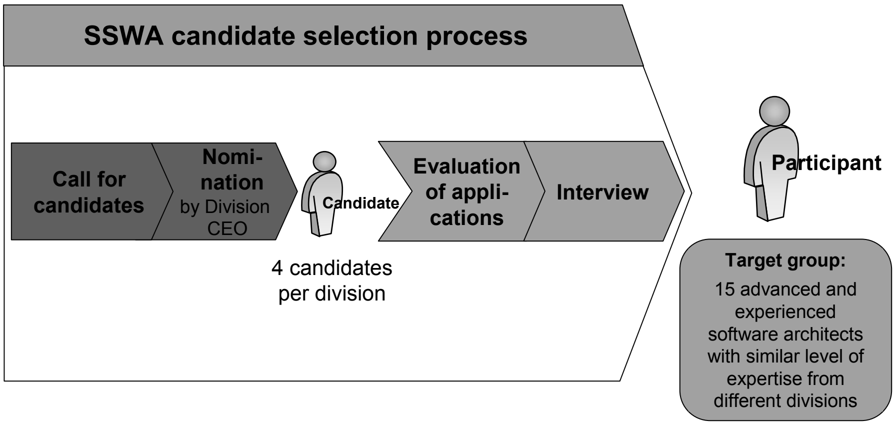
Selected content details: testing and quality


Require-ments Engineering
Software Architecture and Development
Testing and Quality
Business Processes and Strategy
Social Skills and Leadership

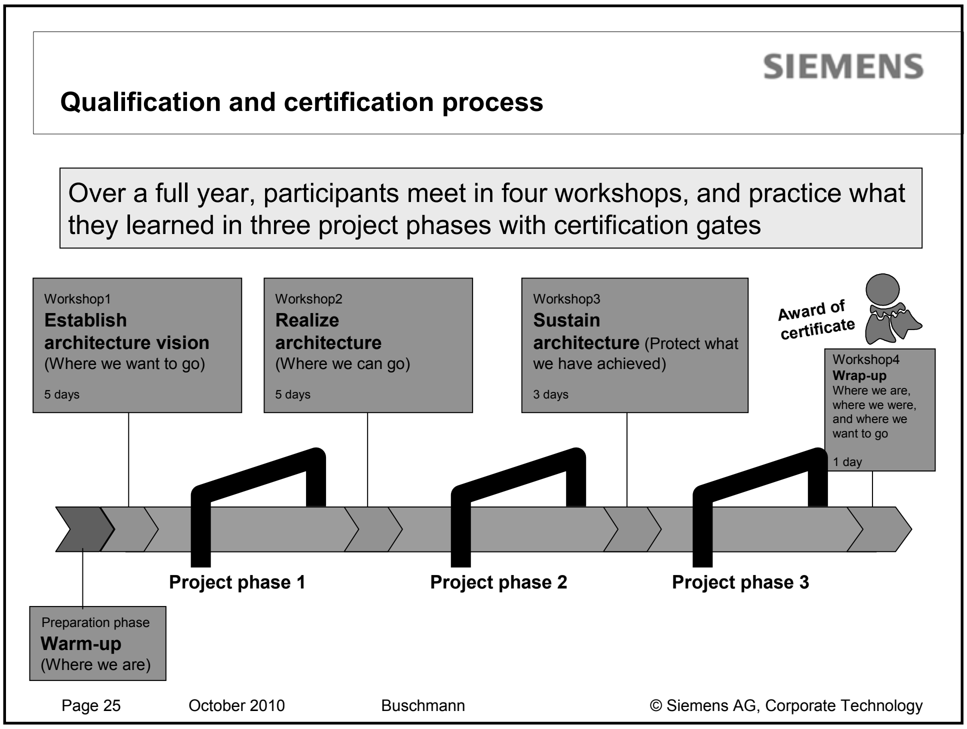
- Understand the mission and the value of testing and promote it
- Risk-based testing strategy
- Test-driven development
- Design for testability
- Test design methods
- Integration testing
- Test architectures, test automation
- Regression testing
- Architectural quality – internal software quality and code quality management

Selection of participants

Curriculum is part of the world-wide employee development program


SSWA candidate selection process
Call for candidates
Nomi-nation by Division CEO
Candidate
Evaluation of appli-cations
Interview
Participant
4 candidates per division

Target group:
15 advanced and experienced software architects with similar level of expertise from different divisions



SIEMENS

Qualification and certification process

Over a full year, participants meet in four workshops, and practice what they learned in three project phases with certification gates

Workshop1
Establish architecture vision
(Where we want to go)
5 days

Workshop2
Realize architecture
(Where we can go)
5 days

Workshop3
Sustain architecture (Protect what we have achieved)
3 days

Award of certificate

Workshop4
Wrap-up
Where we are, where we were, and where we want to go
1 day

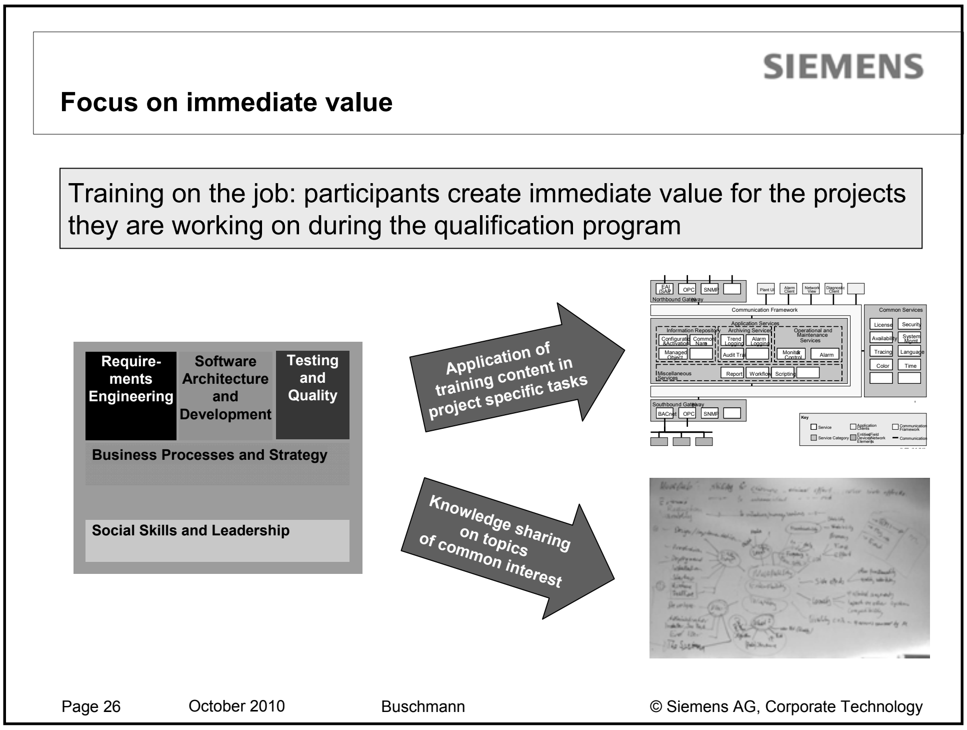
Project phase 1

Project phase 2

Project phase 3

Preparation phase
Warm-up
(Where we are)

Page 25 October 2010 Buschmann © Siemens AG, Corporate Technology



SIEMENS

Focus on immediate value

Training on the job: participants create immediate value for the projects they are working on during the qualification program

Require-ments Engineering	Software Architecture and Development	Testing and Quality
Business Processes and Strategy		
Social Skills and Leadership		

Application of training content in project specific tasks

Knowledge sharing on topics of common interest

Page 26 October 2010 Buschmann © Siemens AG, Corporate Technology

Content

- **Motivation**
- **Software architect target profile**
- **Senior software architect curriculum**
- **Summary**

Experiences and results

Organizations observe a clear improvement of their employees' skills due to participation in the senior software architect qualification program

What do I want to tell Siemens about the program?
Senior Software Architect Qualification Program
Transfer Knowledge
Establish Mindset
encourage Responsibility

100% of the organizations agree that

- the participation was a good investment
- the program enhanced the participants‘ architecture-relevant knowledge

More than 90% agree that

- the contact network of the participant improved
- the program has brought value-add in daily business

Certified senior software architects

- Advise management in software strategy
- Lead architecture teams of high impact projects
- Support critical projects
- Help improve the software engineering maturity of their organizations

SIEMENS

Where we are and want to go

Qualification and certification program for Senior Software Architects is well established and very successful

- Already in the 4th run
- The holistic architecture-driven approach works well for us
- Being able to apply the techniques immediately in current projects is an important success factor, in particular so that the key architects can participate in the qualification program

Qualification and certification program for Software Architects

- Has started in 3 (almost) parallel runs in Europe / USA
- First feedback indicates high acceptance amongst participants

Qualification of Requirements Engineering for Product (Lifecycle) Managers and Project Managers

- Roll out in planning