JAOO conference
- for Developers by Developers

SIEMENS

Corporate Technology

Software Architects and Testing

JAOO Conference 2008
Aarhus, Denmark

Peter Zimmerer
Principal Engineer
Siemens AG, CT SE 1
Corporate Technology
Corporate Research and Technologies
Software & Engineering, Development Techniques
D-81739 Munich, Germany
peter.zimmerer@siemens.com
http://www.siemens.com/research-and-development/
http://www.siemens.com/corporate-technology/

SIEMENS

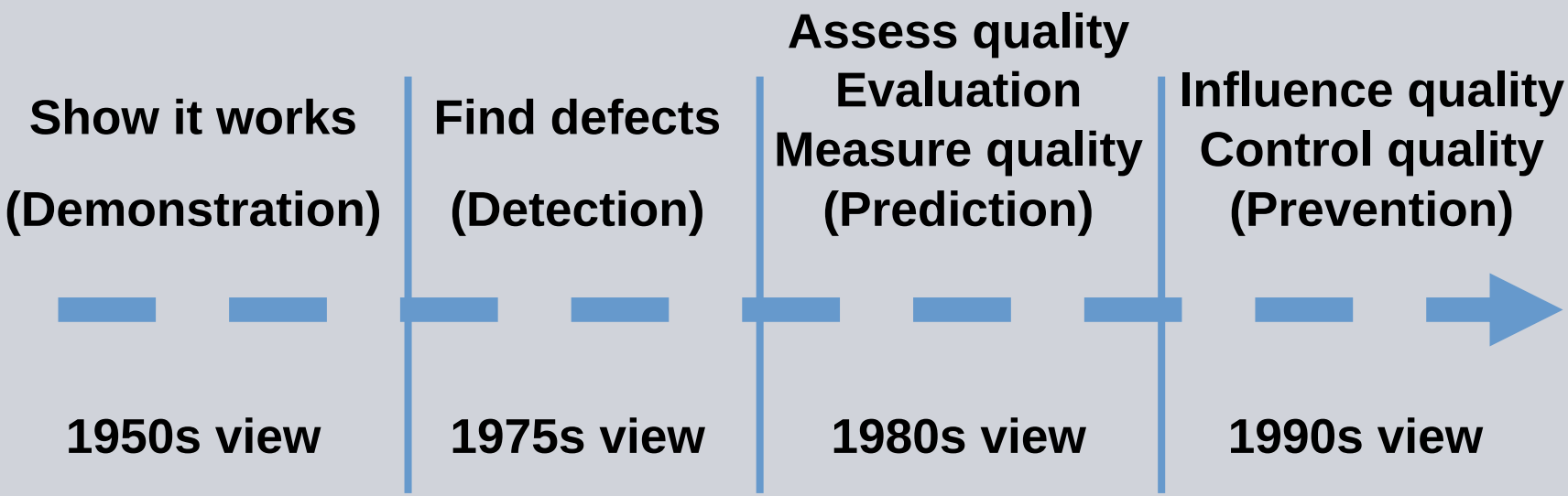
Contents

SIEMENS

Why do we test?

Historical view

History consists of a series of accumulated imaginative inventions.
Voltaire, 1694–1778



Are you sure that you can leave out anything of these?

SIEMENS

Why do we test? What is the *value* of testing?

What is the cost of NOT testing and poor quality?

Empirical technical investigation
of the product / system / artifact / service under test
conducted to provide stakeholders
with information about the quality.

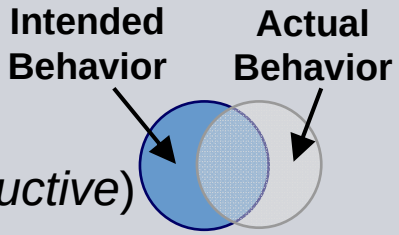
Cem Kaner

Product

Decisions

Process

- **Demonstrate, check:**
  Show it works, gain confidence (*confirmatory, constructive*)
- **Detect, search:**
  Find bugs as early as possible (*investigative, exploratory, destructive*)
- **Mitigate:**
  Reduce risks (*"no risk, no test"*)
- **Assess, evaluate, and predict:**
  Measure quality attributes (for example, performance)
- **Prevent:**
  Control, influence, and drive quality



More / better testing means more / better information!

Some *old* definitions

IEEE Standard 610.12–1990
The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component.

IEEE Standard 829–1998
The process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs) and to evaluate the features of the software item.

BCS SIGIST Testing Standards Working Party
(http://www.testingstandards.co.uk/living_glossary.htm)
The process of exercising software to verify that it satisfies specified requirements and to detect errors.

Some *better* definitions

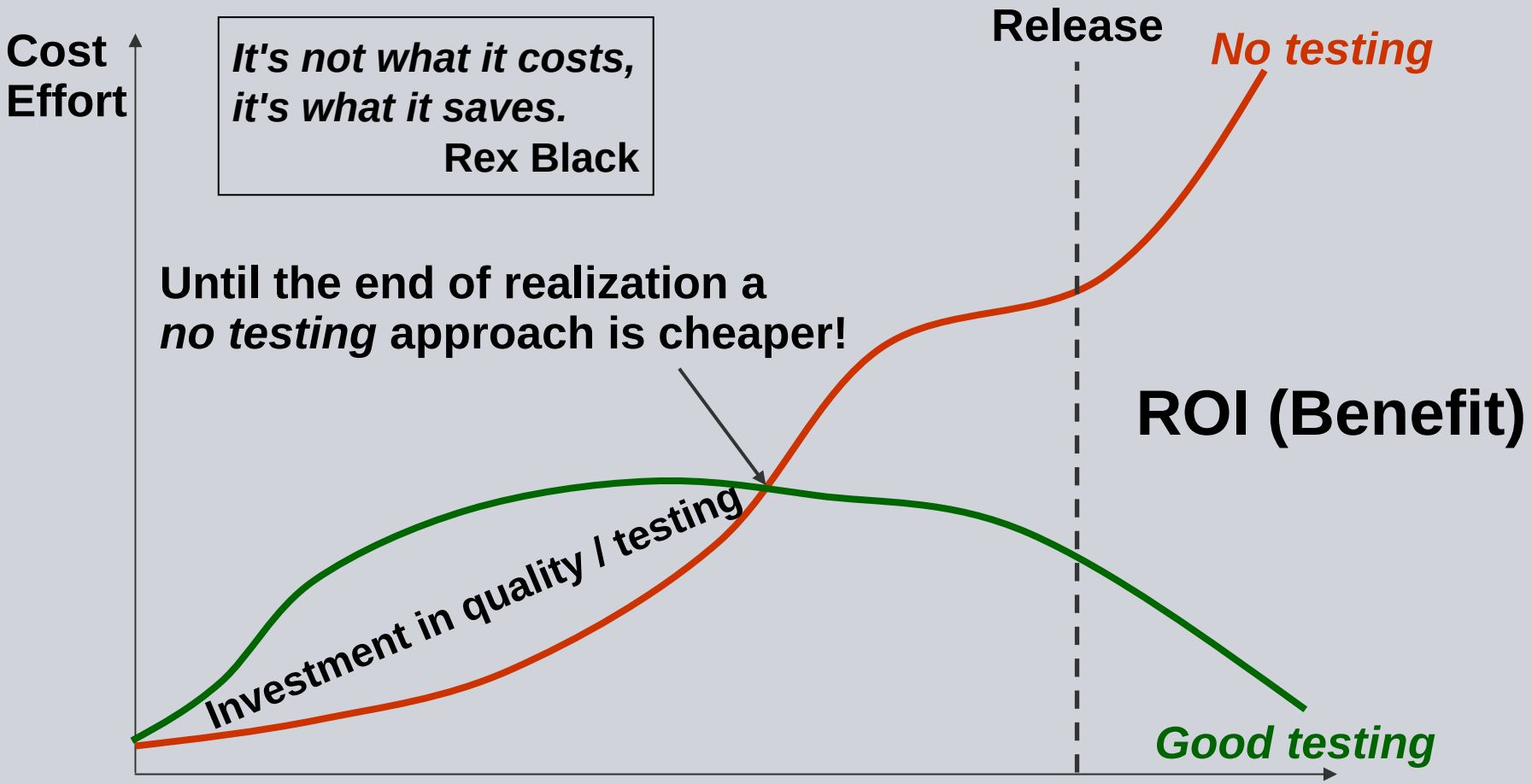
Cem Kaner
Empirical technical investigation of the product / system / service under test conducted to provide stakeholders with information about the quality.

Software Quality Engineering (SQE)
All lifecycle activities concerned with checking software and software-related work products. Testing is an activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Testing includes all activities associated with the planning, preparation, execution, and reporting of tests.

ISTQB Glossary (2007) (http://www.istqb.org/)
The process consisting of all life cycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects.

SIEMENS

Overall cost / effort



This figure is also critical for development of frameworks, platforms, and product lines!

SIEMENS

Testing strategy

Base the testing strategy on business goals and priorities

Ingredients of a testing strategy:
What to test? Where? Why? When? Who? How (much)?
Test basis, system composition, risks, test levels, entry and exit criteria, test effort distribution, test techniques, test automation, regression testing, test process, defect management, and so on

Typically the testing strategy is documented in a Master Test Plan

No risk – no test
Risk = Probability of failure × Damage

Project risks – Process risks – *Product risks* (focus in testing)
→ Product risk analysis

Use a tabular form to develop and define the testing strategy

SIEMENS

Product risk analysis for software architects

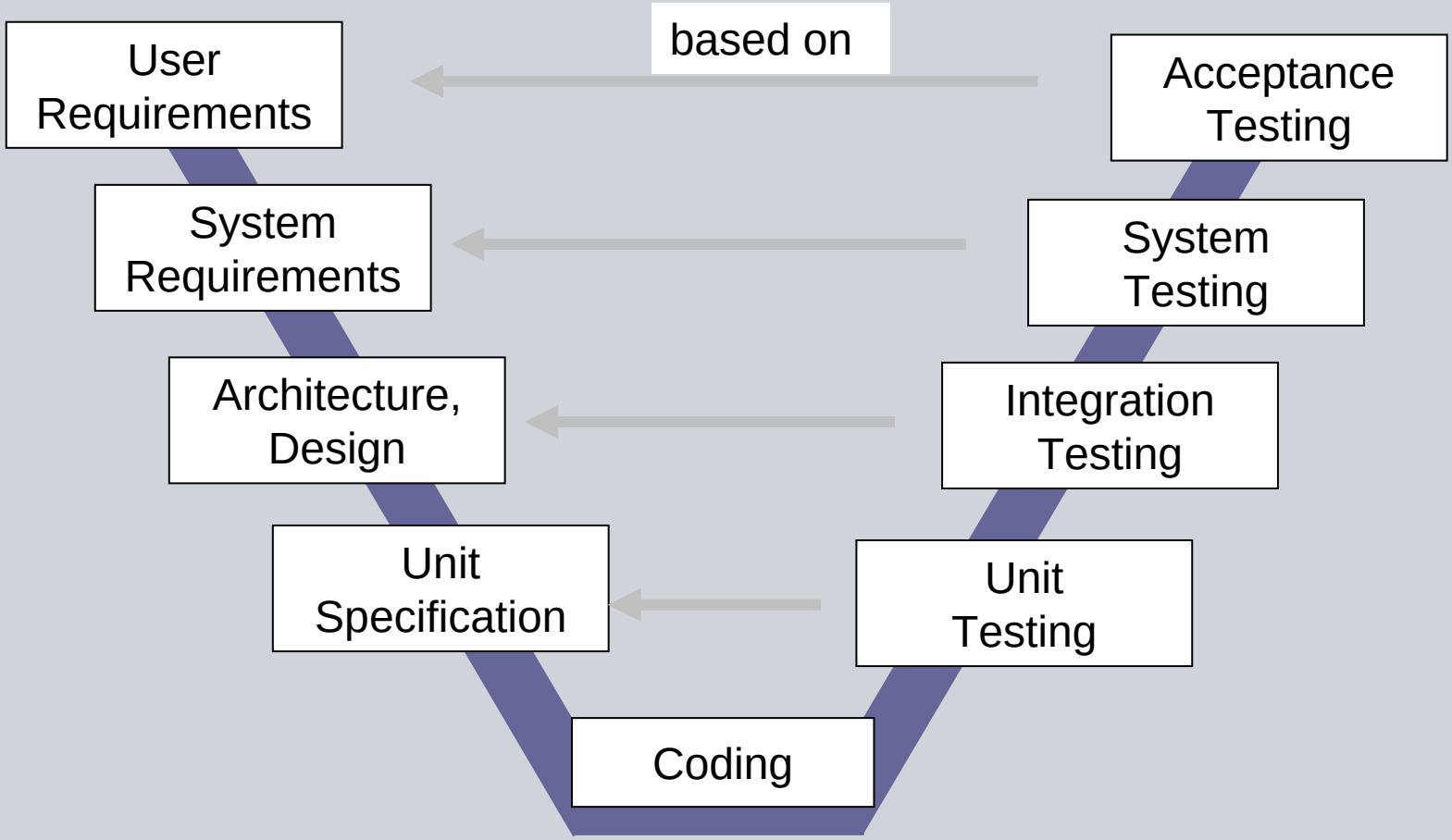
Identify especially all risks related to architecture

- Architectural requirements (functional and non-functional)
- Architectural quality attributes
- Architectural use cases, features, and functions
- Architectural decisions
- Design decisions
- Technology selections
- Third-party component selections (open source)
- Bug categories

Actively participate in a product risk analysis workshop as one important stakeholder

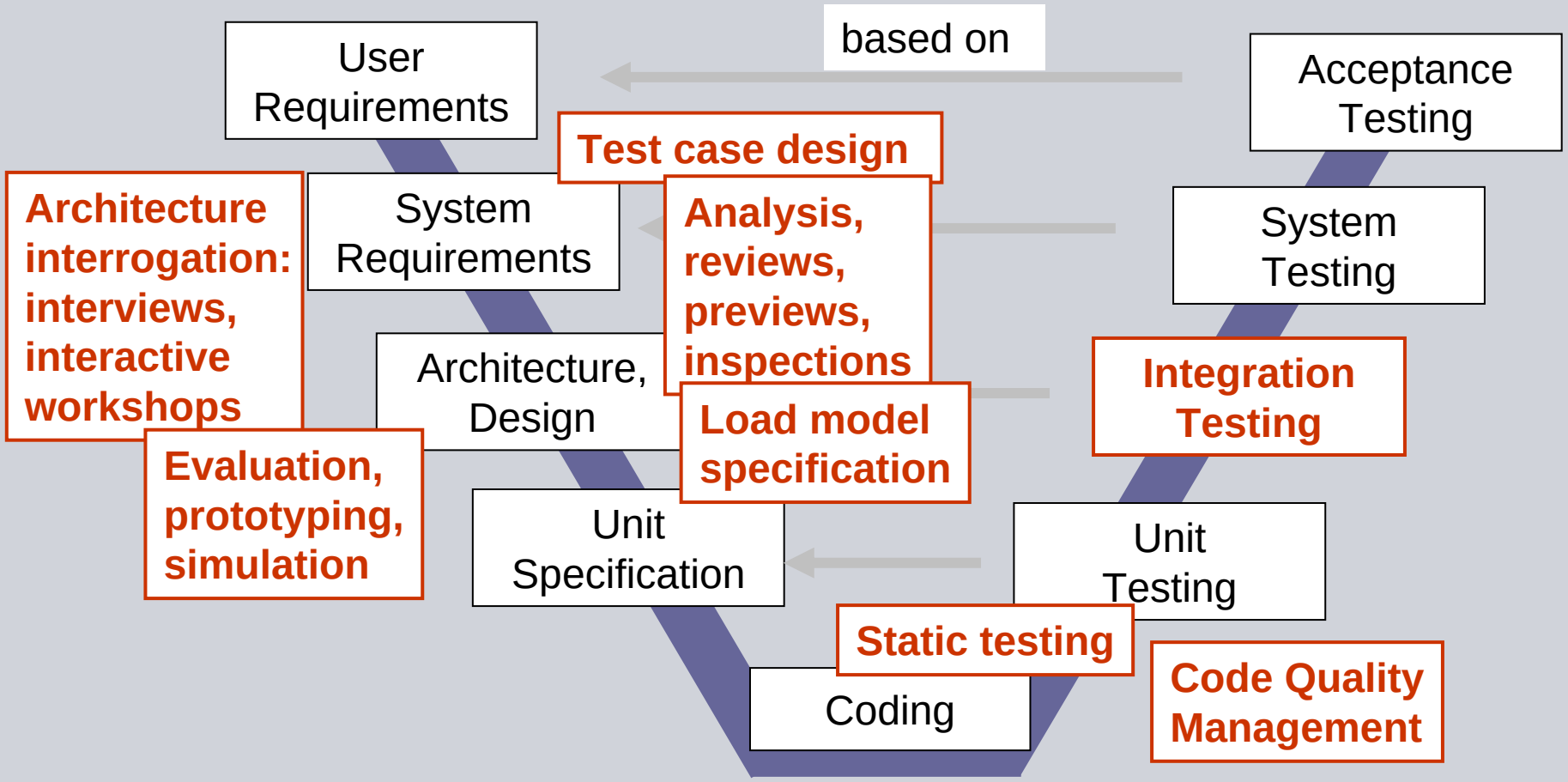
SIEMENS

Test levels – example V model



SIEMENS

Test levels – example V model with architecture testing



Architecture testing

– any testing of architecture and architectural artifacts

Architecture testing @ Google

	Result hits @ Google:
Requirements testing	199,000
Architecture testing	24,000
Document testing	25,000
Unit testing	1,590,000
Integration testing	721,000
System testing	1,760,000
Acceptance testing	1,130,000

SIEMENS

Test-driven development (TDD) ≈ Preventive Testing

Preventive testing is built upon the observation that one of the most effective ways of specifying something is to describe (in detail) how you would accept (test) it if someone gave it to you.

David Gelperin, Bill Hetzel (<1990)

Given any kind of specification for a product, the first thing to develop isn't the product, but how you'd test the product. Don’t start to build a product till you know how to test it.

Tom Gilb

The act of designing tests is one of the most effective bug preventers known.

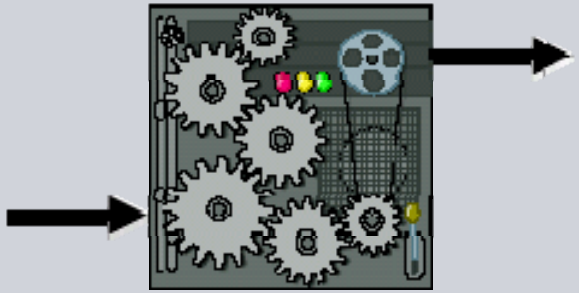
Boris Beizer, 1983

SIEMENS

Design for testability

Visibility / observability

- *What you see is what you test*
- Ability to observe the outputs, states, internals, resource usage, and other side effects of the software under test
- Interaction with the system under test through observation points



Control(lability)

- *The better we can control the software, the more testing can be automated and optimized*
- Ability to apply inputs to the software under test or place it in specified states (for example reset to start state)
- Interaction with the system under test through control points

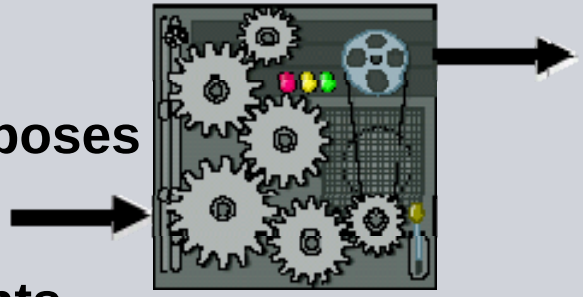
SIEMENS

Design for testability – How?

Suitable testing architecture

Additional (scriptable) interfaces for testing purposes

Interaction with the system under test through well-defined observation points and control points



Coding guidelines, naming conventions

Built-in self-test

Visibility / observability

Control(lability)

Consistency checks (assertions, design by contract)

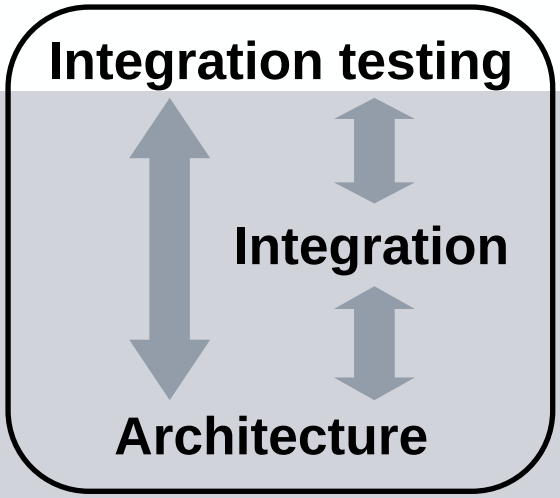
Logging and tracing

Diagnosis and dump utilities (internal states)

Think test-first: how can I test this?

Integration testing

SIEMENS



An integrated system is a complex equation, built from interdependent variables. Explore these variables and their interactions.

Len DiMaggio, 2003

The goal of integration testing is to test in a grey-box manner

- The interaction of components and subsystems
- The interaction and embedding with the environment and system configuration
- The dynamic behavior and communication of the system
- Control flow and data flow
- The architecture and design as specified in the **S**oftware **A**rchitecture **D**escription document

During integration test execution the internal behavior of the system under test is monitored by using tracing facilities to provide the required information

SIEMENS

Test architectures, test automation

The architecture of a test system may be even more complex than the architecture of the system under test

Use architecture best practices to define, specify, create, realize, and maintain the test system as well

Automated testing is the foundation for any kind of iterative or agile development

- Daily builds and small releases are useless if they cannot be validated
- But carefully decide on start / extent of test automation implementation

An efficient quality check of software is not possible in real project life without the *right* test automation!

- Incremental, iterative, agile processes
- Regression testing
- Maintenance – refactoring, redesign, reengineering

SIEMENS

Regression testing – The big questions are always …

Software Architect:
Which parts of the architecture must be tested again after a (code) change?

Test Manager:
Which test cases must be executed again after a (code) change?

Remark:
Change can be a *code change*
but also *any change in the environment or system configuration*.

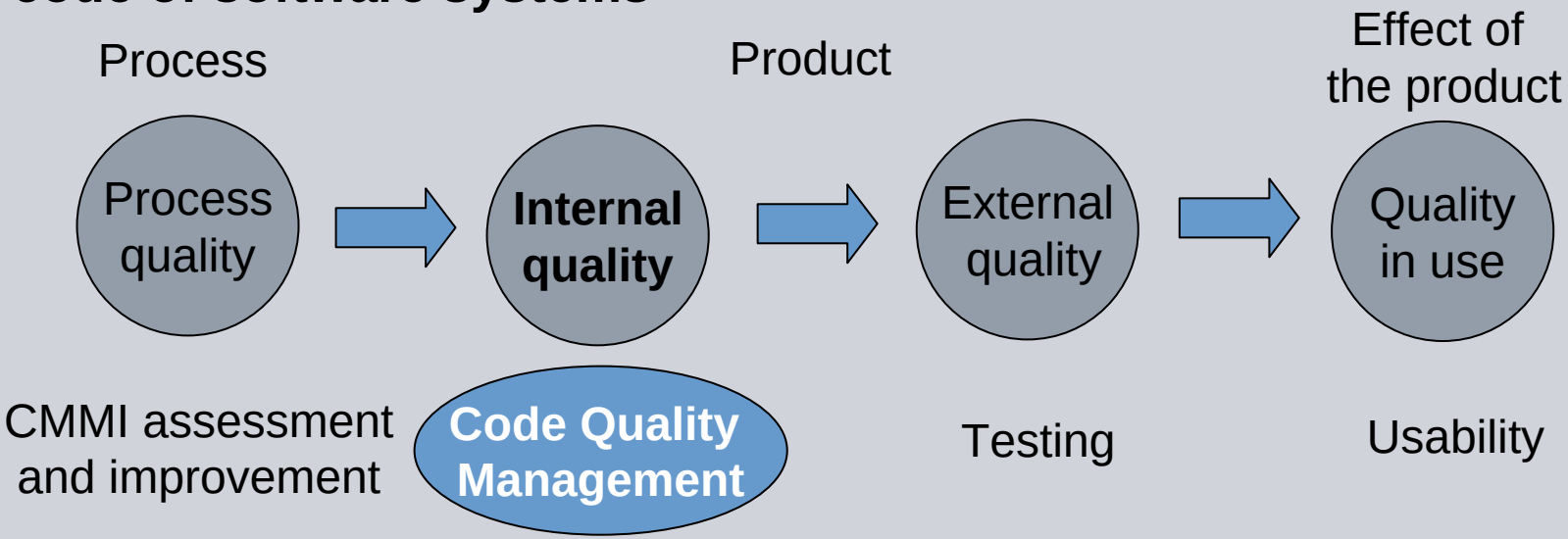
Use your knowledge about the architecture and design of the system during change and impact analysis

- to identify risky areas
- to select required regression test cases effectively

Architectural quality – internal software quality and code quality management

SIEMENS

Code Quality Management (CQM) assures the fitness of the source code of software systems



The internal quality, or code quality, significantly affects the external quality and the effort required for maintenance and extension of a software system

⇨ Code Quality Management (CQM) is an important lever for improving the overall quality of a software system

SIEMENS

Tasks and benefits of code quality management

Architectural conformance checking

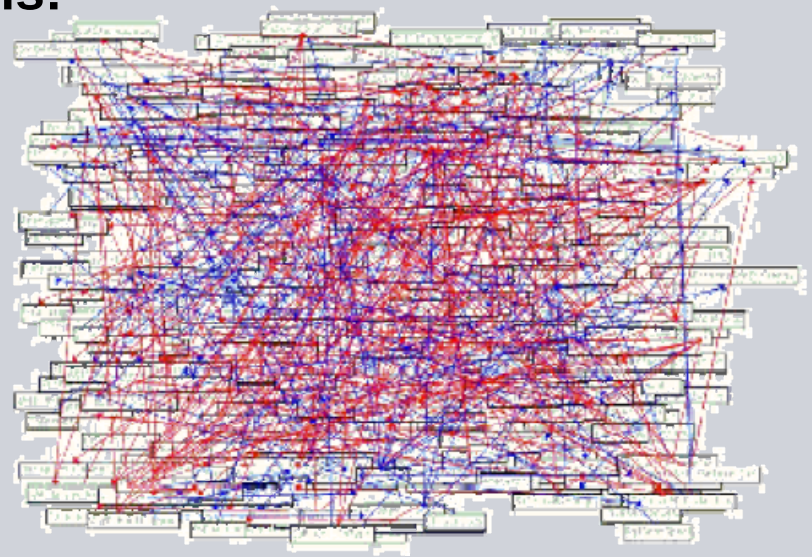
- Provide information and deliver evidence that the architecture is implemented as specified

Identify suspicious components and trends (software aging)

Impact analysis and code comprehension

- Speed up development by facilitating change
- Decrease extension and maintenance efforts
- Enhance reusability of components

Avoid this:



SIEMENS

Involvement of Software Architects in testing activities

Understand the mission and the value of testing and promote it

Risk-based testing strategy

Test-driven development

Design for testability

Integration testing

Test architectures, test automation

Regression testing

Architectural quality – internal software quality and code quality management

Summary

The Software Architect cooperates closely with the Test Manager to define, motivate, drive, and enforce a comprehensive understanding of and attitude to testing and quality in the whole team

The Software Architect and the Test Manager address "quality" from different viewpoints:

- Software Architect:
  Build quality into the system (or rather into the architecture)
- Test Manager:
  Provide information related to quality (feedback)

SIEMENS

A departing thought

If you want to build a high-quality software architecture,
don't drum up people to
collect requirements and design specifications,
divide the testing work,
and give orders.
Instead, teach them to yearn for
a sustainable architecture with quality inside
and satisfied clients.

Freely adapted from Antoine de Saint-Exupéry