FOKUS

Fraunhofer Institute for Open
Communication Systems

Competence Center
Modelling and Testing

MOTION

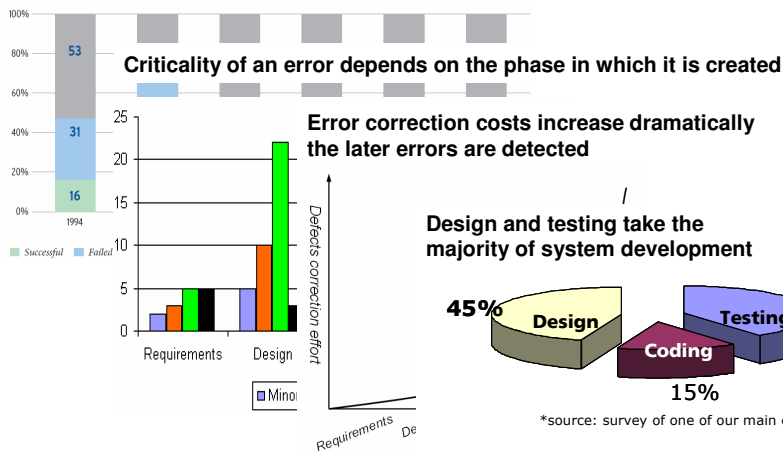# Modelling und Testing:

Model-Based Testing

Ina Schieferdecker
ina.schieferdecker@fokus.fraunhofer.de
+49 30 3463 7241

Outline

- Motivation
- Principles of Modell-Based Testing
- An Example Process
- Summary

Current Situation for Software-Based Systems

Number of successful software projects is decreasing

Criticality of an error depends on the phase in which it is created

Error correction costs increase dramatically the later errors are detected

Design and testing take the majority of system development

45% **Design**   **Testing** 40%

Coding

15%

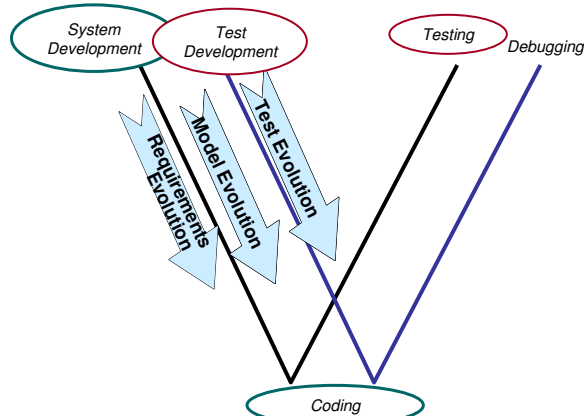*source: survey of one of our main customers*

System and Test system

Although programmers, testers and program managers know that code must be designed and tested, many appear to be unaware that tests themselves must be designed and tested –

by a process no less rigorous and no less controlled than that used for code.

Boris Beizer

The W-Model

- Reflected in the V-Modell XT by check lists

- No explicit requirement for such checks – a design issue
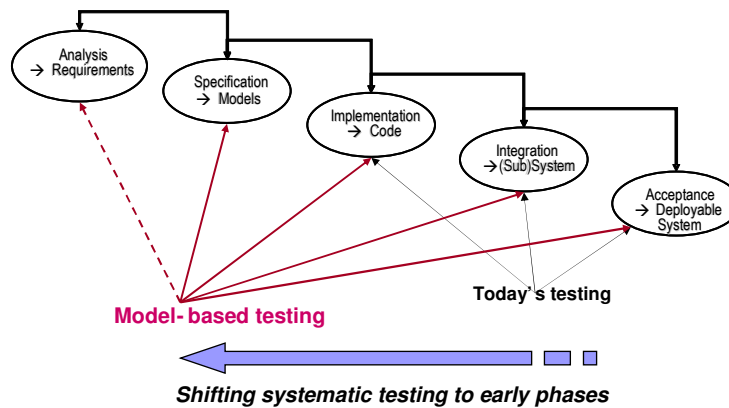
- Model-based testing as a way forward

Model-Based Testing

- ... refers to <u>software testing</u> where <u>test cases are derived in whole or in part from a model</u> that describes selected, often <u>structural, functional, sometimes non-functional aspects</u> of a system under test (SUT). [Wikipedia, 2007]
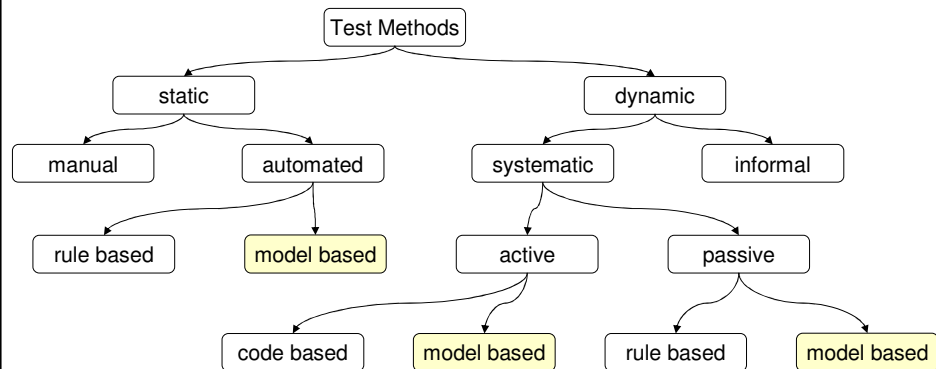

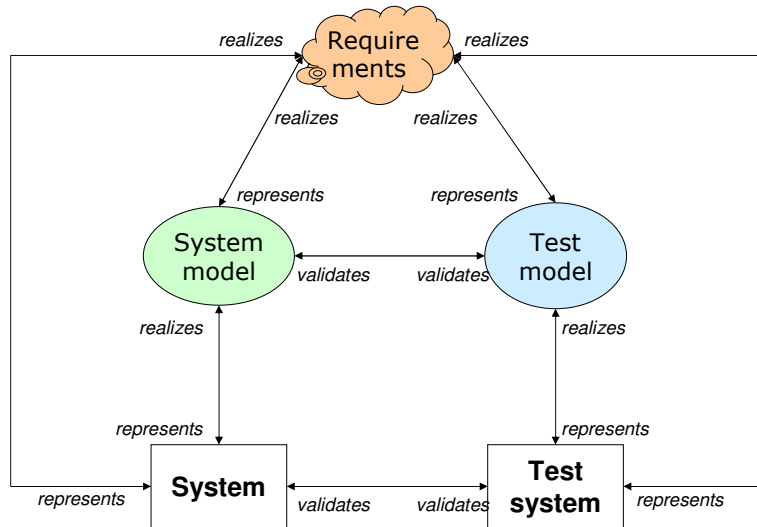**Shifting systematic testing to early phases**

3

Model-Based Testing (2nd)

- Various system models exist such as:
    - Requirements models
    - Information models
    - Work flow models
    - Architectural models
    - Behavioural models
    - Configuration models
    - Deployment models
    - Performance models
    - Risk models
    - …

- hence, model-based test methods differ in the <u>system model</u> being considered, the methods taken for <u>test generation</u> and the way <u>test results</u> are being obtained
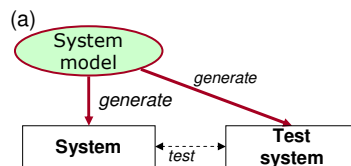
MBT Placement in Testing

```
                        Test Methods
                    /                 \
                static              dynamic
               /      \            /        \
          manual    automated  systematic   informal
                    /      \      /      \
            rule based  model based  active   passive
                              /    \      /    \
                    code based  model based  rule based  model based
```

4

Principal Approach

Different Forms

System-driven
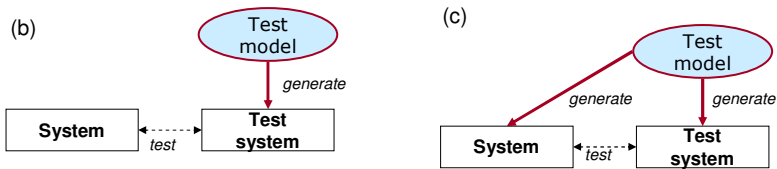
(a)

Test-driven

(b)

(c)

Combined models

(d)

(e)
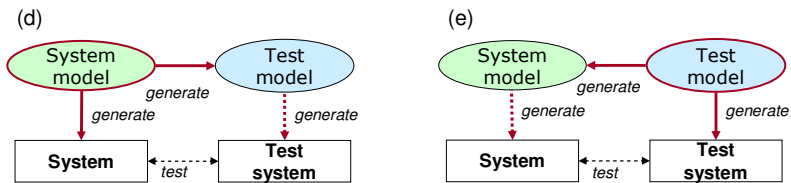
Combined, but independent models

(f)

6
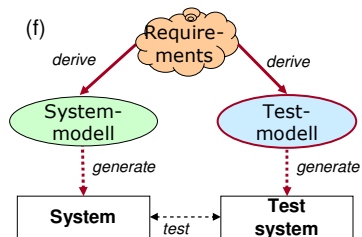
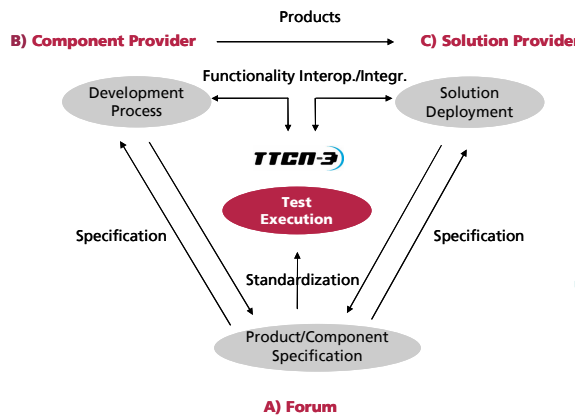| UML diagram	Test Level				Test Kind		Test Method	
	Unit	Component	Integration	System	functional	non-functional	static	dynamic
Structure								
class	✗	✗	✗	✗	✗		✗	✗
composite structure		✗	✗		✗		✗	✗
component		✗	✗	✗	✗		✗	✗
deployment			✗	✗	✗		✗	
object		✗	✗		✗		✗	✗
package			✗		✗		✗	
Behavior								
use case			✗	✗	✗	✗[1]	✗	✗
activity			✗	✗	✗	✗[1]	✗	✗
sequence			✗	✗	✗	✗[1]	✗	✗
communication		✗	✗		✗	✗[1]	✗	✗
interaction overview			✗	✗	✗	✗[1]	✗	✗
timing		✗	✗			✗[1]	✗	✗
state machine	✗	✗	✗	✗	✗	✗[1]	✗	✗

1 By use of additional profiles

Test Specification Techniques

- The Testing and Test Control Notation **TTCN-3**

- A standardized alternative to proprietary test systems
    - Developed by a large group of testing experts
    - Used by a growing community
    - Proven by tools
    - Maintained at ETSI

- A test specification and implementation language
    - A multipart standard covering
        - textual TTCN-3 core
        - graphical TTCN-3
        - execution interfaces TRI and TCI
        - language mappings to TTCN-3, e.g. for IDL

Test Specification Techniques (2nd)

- TTCN-3: one test technology for different tests
  - Distributed, platform-independent testing
  - Integrated graphical test development, documentation and analysis
  - Adaptable, open test environment

Products

B) Component Provider → **C) Solution Provider**

Functionality Interop./Integr.

Development Process ← → Solution Deployment

TTCN-3

Test Execution

Specification   **Specification**

Standardization

Product/Component Specification

A) Forum

- **The testing middleware**
  - unifying the documentation and definition of tests
  - unifying the tests in IT, Internet- and Telco-based systems (supporting their convergence)
  - unifying the test infrastructure

- **Areas of Testing**
  - Regression Testing
  - Conformance/Functionality Testing
  - Interoperability/Integration Testing
  - Load/ Stress Testing

Test Specification Techniques (3rd)

- Test design with the UML Testing Profile
  - Integrated into UML based development processes

- U2TP has been defined to capture all information that would be needed by different test processes
- It is a testing profile based upon UML 2.0
  - That enables the test definition and test generation based on structural (static) and behavioral (dynamic) aspects of UML models, and
  - That is capable of inter-operation with existing test technologies for black-box testing

- It is an official OMG standard since summer 2005

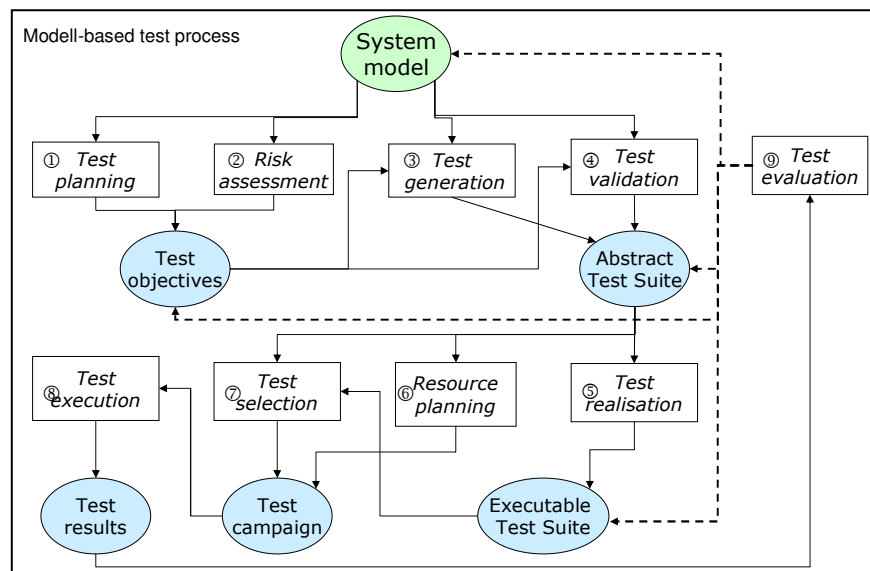## Test Specification Techniques (4th)

	U2TP	TTCN-3
Test design	✓	(—)
Test specification	✓	✓
Test execution	(—)	✓
Format	Graphical	Textual and graphical
Transformation	U2TP to TTCN-3 (✓)	TTCN-3 to U2TP ✓

The Overall Picture

Modell-based test process

① Test planning
② Risk assessment
③ Test generation
④ Test validation
⑨ Test evaluation
Test objectives
Abstract Test Suite
⑧ Test execution
⑦ Test selection
⑥ Resource planning
⑤ Test realisation
System model
Test results
Test campaign
Executable Test Suite

9

OMG MDA: A Model-Centric System Development

CIM
Computation Independent System Models

PIM
Platform Independent System Models

PSM
Platform Specific System Models

System Code

Specify Systems/ Applications

QVT
Specify Transformations

Specify Languages
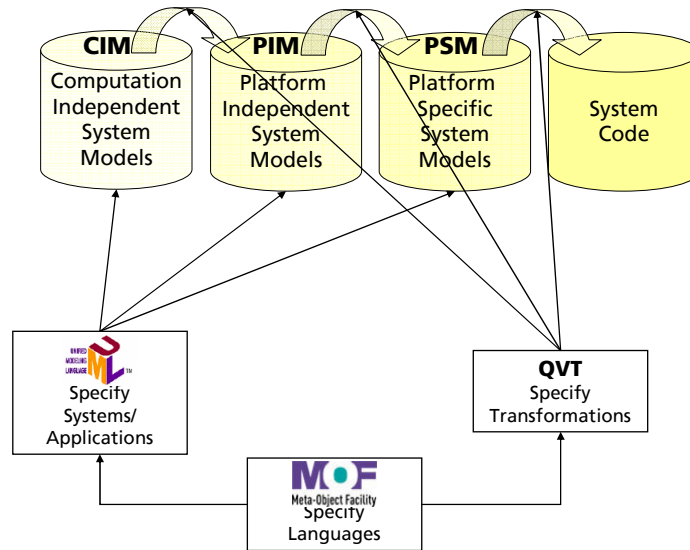
MDA Extended: Integrated Modelling and Testing

Business concepts — *derive* — Require-ments — *derive* — Test concepts

Business Modell   CIM — *generate / test* — CIT

Plattform independent IT model   PIM — *generate / test* — PIT

Plattform dependent IT model   System platform   PSM — *generate / test* — PST   Test platform

System — *test* — Test-system

CIM   Computation Independent Model
CIT   ... Test
PIM   Platform Independent Model
PIT   ... Test
PSM   Platform Specific Model
PST   ... Test
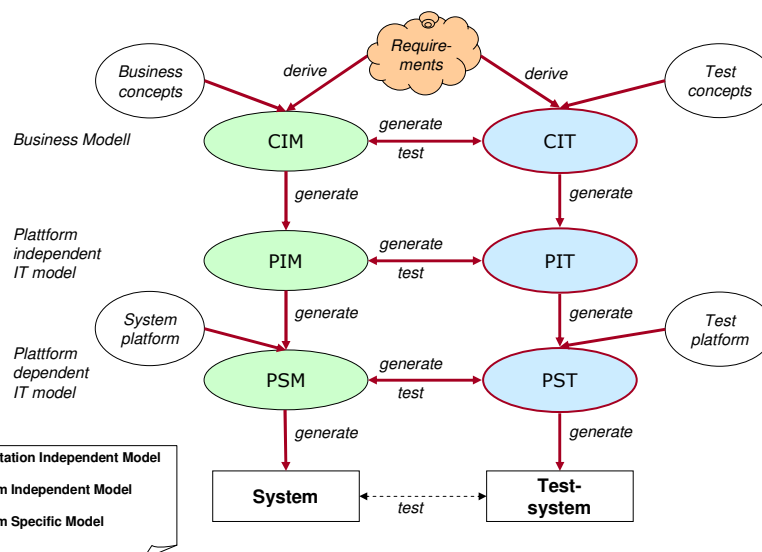
Detailed MDA Extended

Used for Japanese IT applications
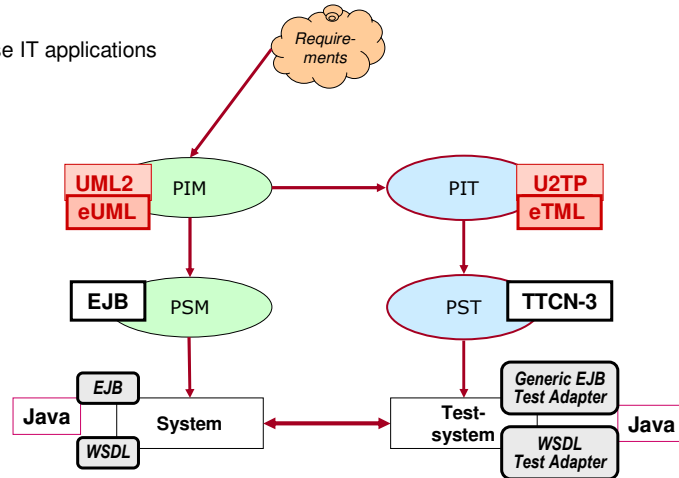
Underlying Technologies
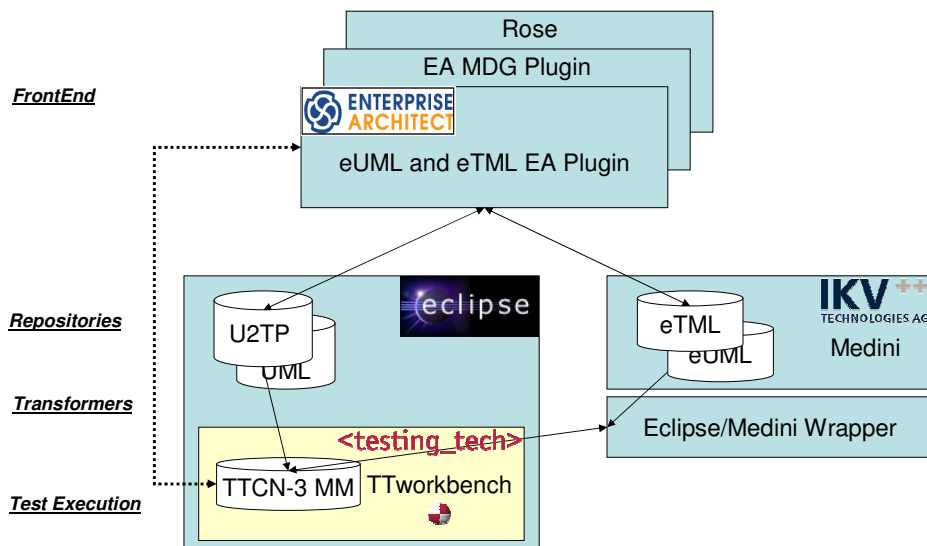
11

A Small Example

d.   ...
e.   Derive test configurations
f.   Derive interfaces
g.   Derive test behavior
h.   ...
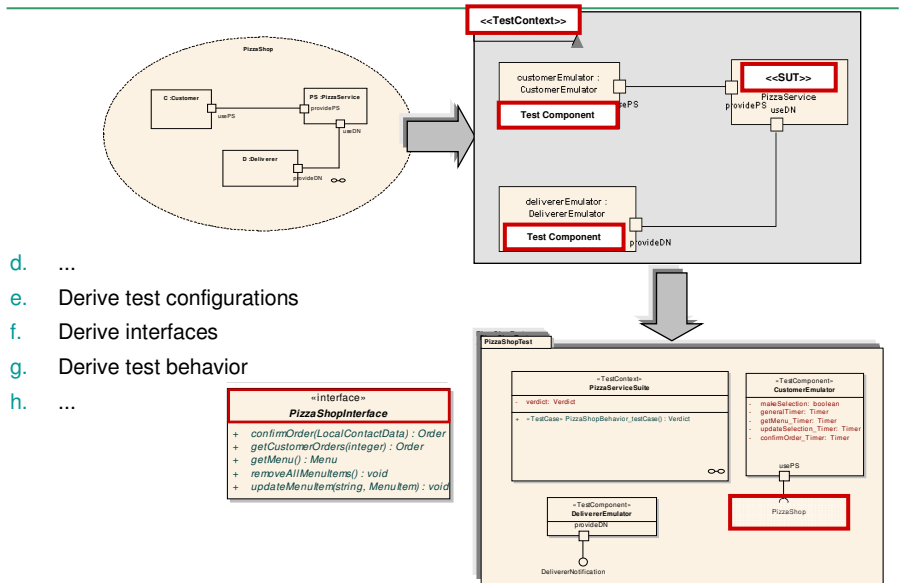
Run the Tests and Evaluate the Test Results
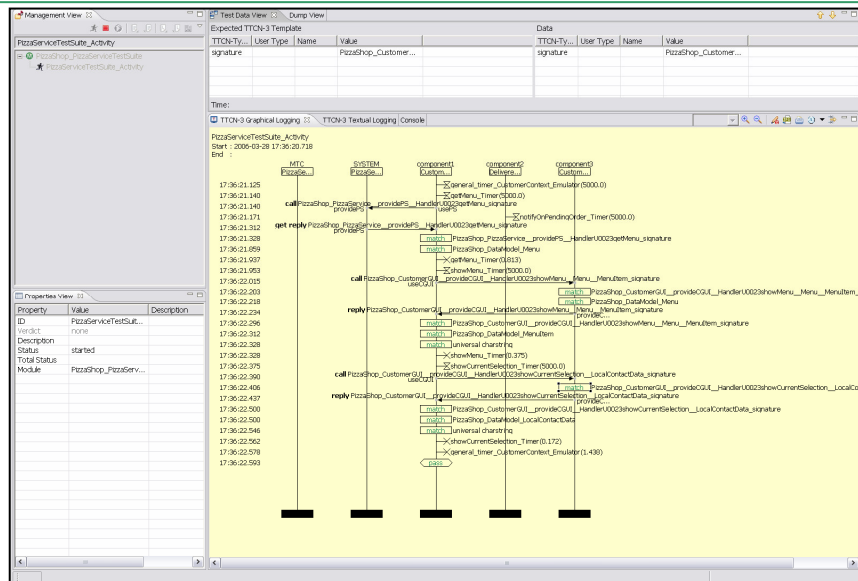
Summary

- Basic idea of model based testing approach:
  Testing complex software by means of
  exhaustive coverage of relatively simple models
- Area of applicability:
  Any software and hardware components with
  well-defined interfaces or functional properties

- Formalization of requirements allows
  improving most phases of software development processes
- Formalized methods should be
  well integrated in existence tools and processes

- Tools:
  - Numerous TTCN-3 tools, some U2TP tools
  - Selected test generation tools

MBT means also Test Models: TTCN-3 and U2TP

- Both are
  - test specification techniques
  - open to various domains
  - open to various development and test processes
- TTCN-3 is widely supported by tools, U2TP is emerging
- U2TP enables test design along UML based design processes

- TTCN-3 is the choice for technical testing and automated execution
- U2TP is the choice for high-level test design and modelling

- Both are a basis for systematic testing and can be integrated into the
  system development

Improvement Potentials

- Well-defined test processes in place
- Emphasis on integration and system tests
- Test development and specification like in system development
- Test development as early as possible and as continuous as possible
- Test automation in order to concentrate on the what to be tested and not the how

- Test generation
  - Testability of models
  - Test hypotheses, heuristics for efficient and effective test generation
  - Adequate reflection of test data in test generation
  - Definition and use of test patterns
- Test quality measures
- Traceability from requirements via models to tests, test campaigns and system bugs
- ...

Thank you !

Any further questions?