Fraunhofer

FOKUS

RCIS 2013 Tutorial

Test automation with models

Marc-Florian Wendland, Ina Schieferdecker, | RCIS 2013 | 30th May, 2013 | Paris, France

Goal of this tutorial

- Provide insights into the principles of software test automation
- Provide an overview of the state of the art in industrial test automation
- Stress out why test automation with models can alleviate challenges in testing
  - No discussion about test generation algorithms or modeling for test case generation
- Differentiate the different kind of models participating in model-based testing approaches
- Provide an overview of most recent standardization activities with regards to model-based testing
- Summarizes key findings from industrial application of model-based testing

Agenda

- Introduction
- Test automation with models
- Industrial standards and notations
- Findings from industry
- Conclusion and discussion

Agenda

- **Introduction**
- Test automation with models
- Industrial standards and notations
- Findings from industry
- Conclusion and discussion

Introduction
What is automation in general?

Automation is the use of machines, control systems and information technologies to optimize productivity in the production of goods and delivery of services

Advantages

- Increased throughput or productivity.
- Improved quality or increased predictability of quality.
- Improved robustness (consistency), of processes or product.
- Increased consistency of output.
- Reduced direct human labor costs and expenses.
- Repeatability with remaining precision

Disadvantages

- Security Threats/Vulnerability
- Unpredictable/excessive development costs
- High initial cost
- Clear process structures

Source: [Wik13]

Introduction
Distinguish between intellectual and clerical tasks

Requires expert knowledge

Manual clerical task

Requires expert knowledge

Intellectual task

Requires expert knowledge

Automated clerical task

Productivity of the machine for 1 hectar (about 3h – 5h)

compared to

40 – 60 people

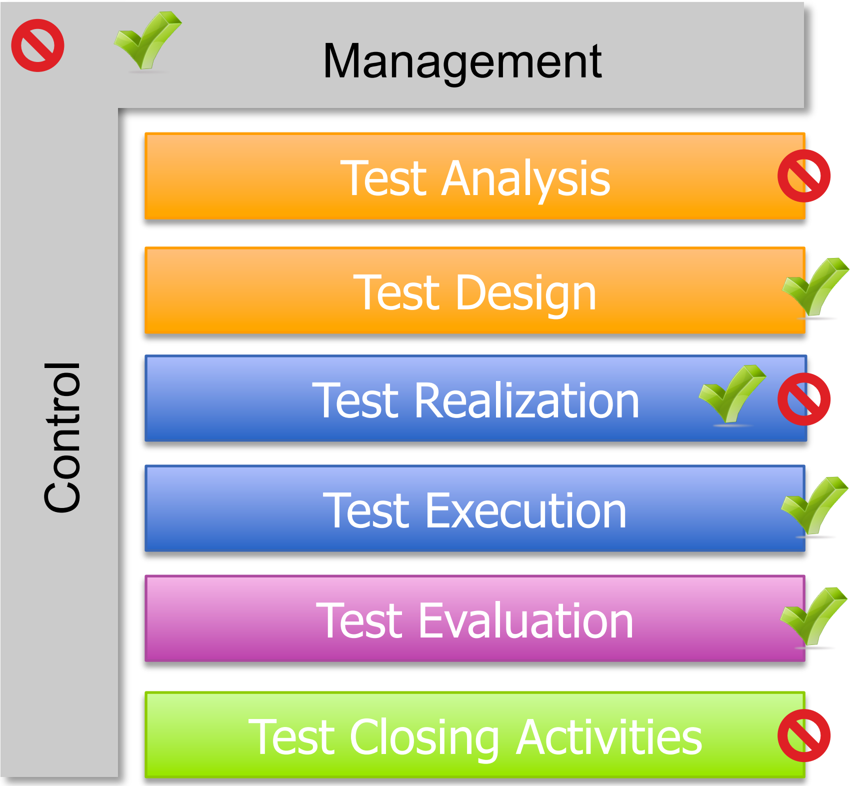
Introduction
What is software test automation?

The use of software to perform
or support
test activities,
e.g., test
management, test design, test execution and results checking.

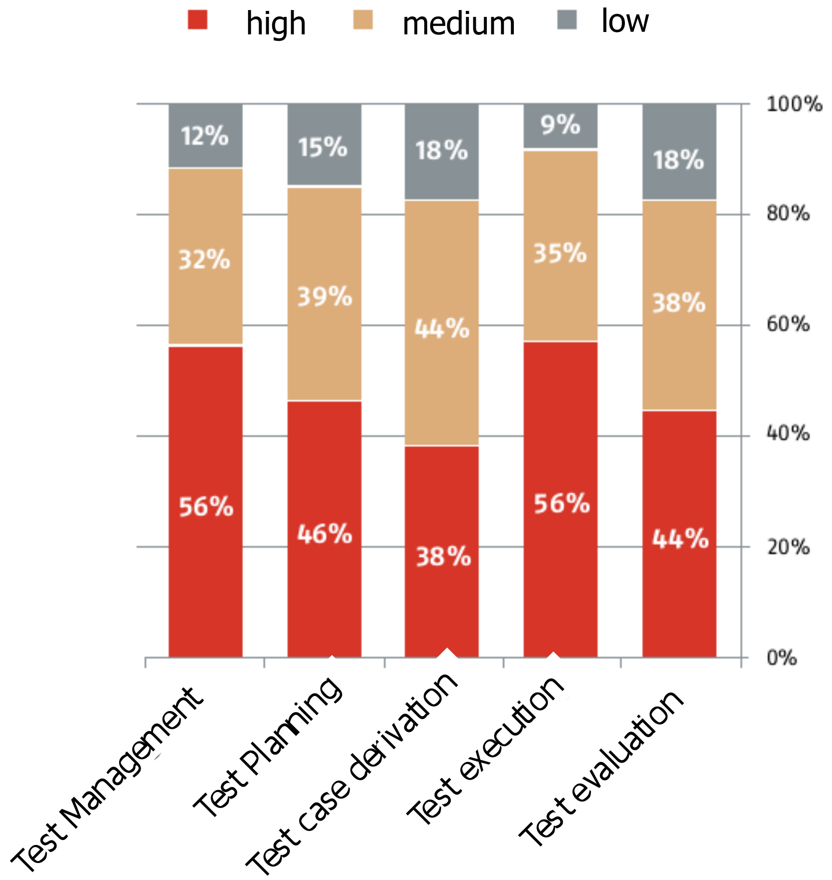
Source: [ISTQB]

The use of software to perform test activities
in an automated way
such as
test scheduling, test design, test execution, test evaluation etc.

Introduction
Test automation of test process activities



Satisfaction with testing activities



[SwissQ]

Introduction
Test automation in Industry

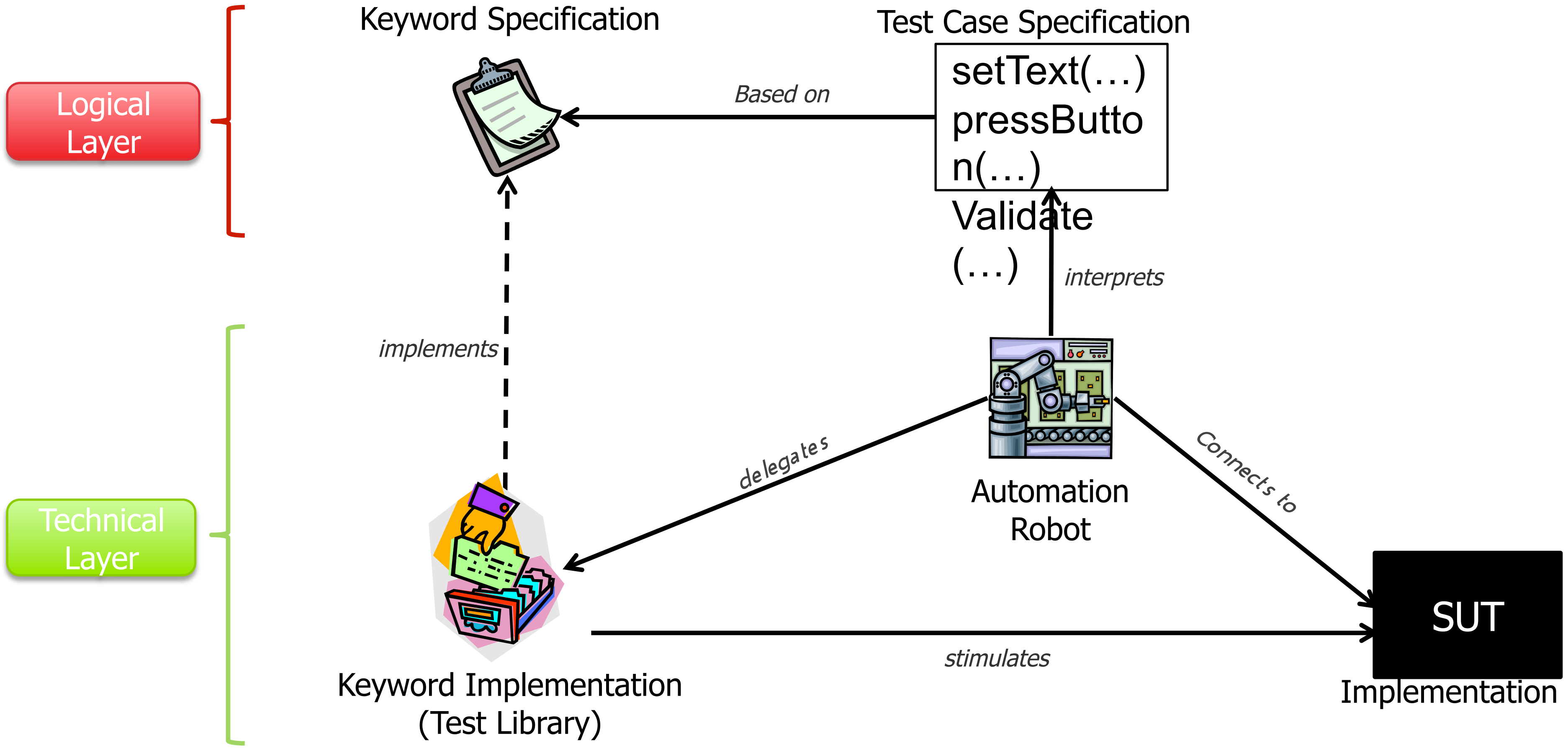
State of the Practice

Automated Test Execution

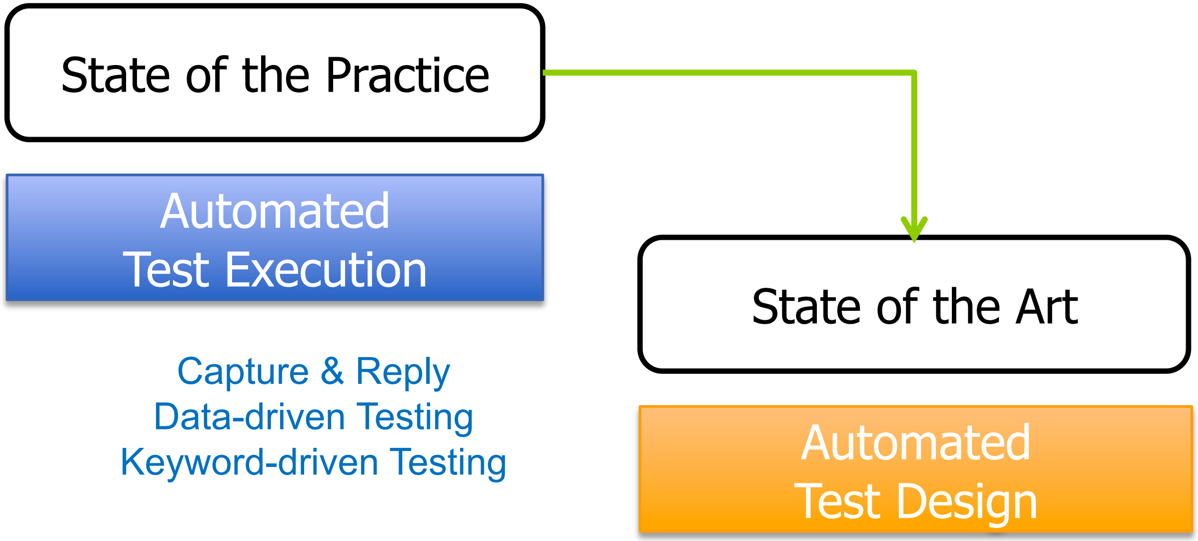
Capture & Reply
Data-driven Testing
Keyword-driven Testing

Introduction

State of the Art in automated test execution - Keyword-driven testing



Introduction
Test automation in Industry



State of the Practice

Automated Test Execution

Capture & Reply
Data-driven Testing
Keyword-driven Testing

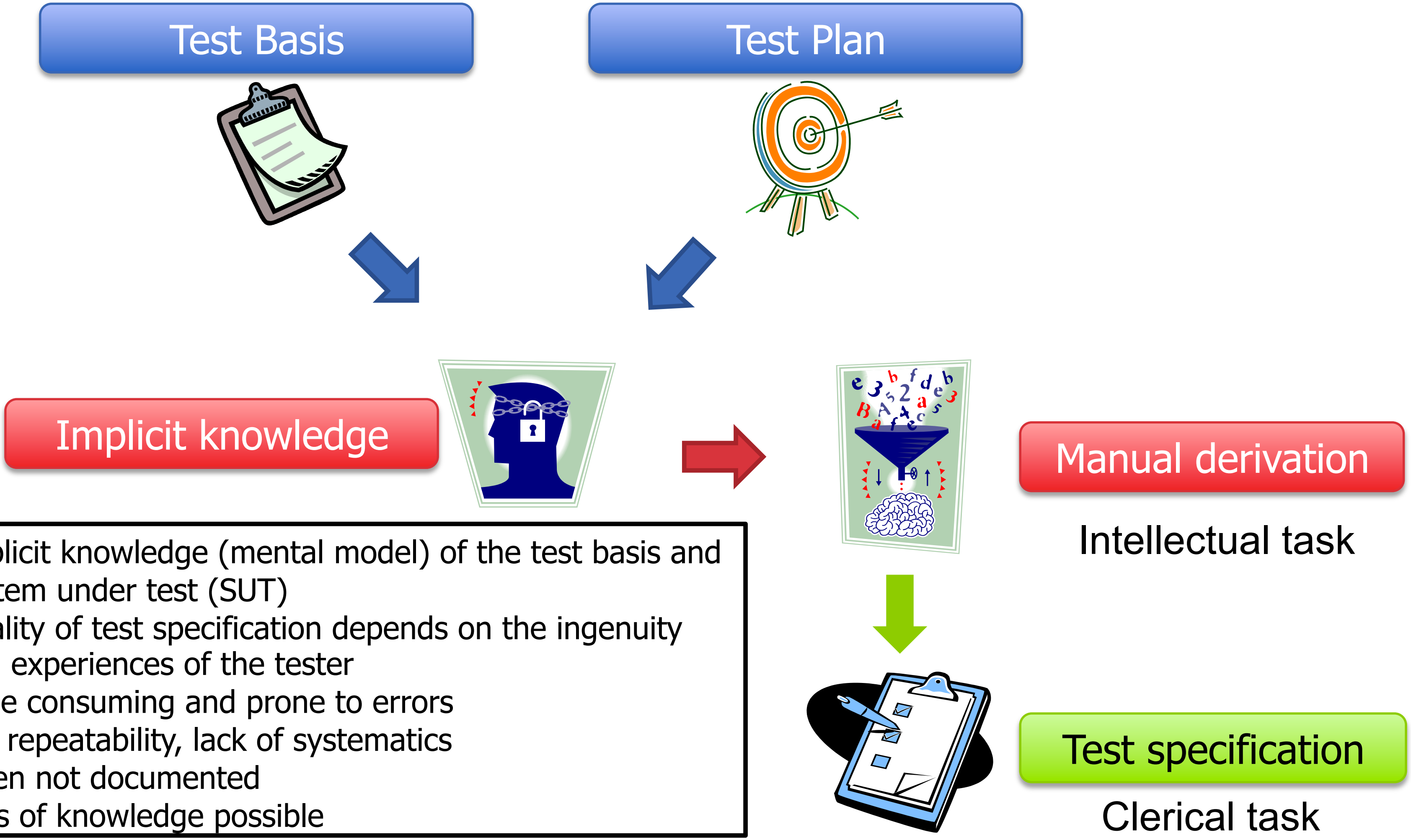
State of the Art

Automated Test Design

Incremental Test case generation
Test data generation
Test script generation

Introduction

State of the art in test design – Traditional testing



Test Basis

Test Plan

Implicit knowledge

Manual derivation

Intellectual task

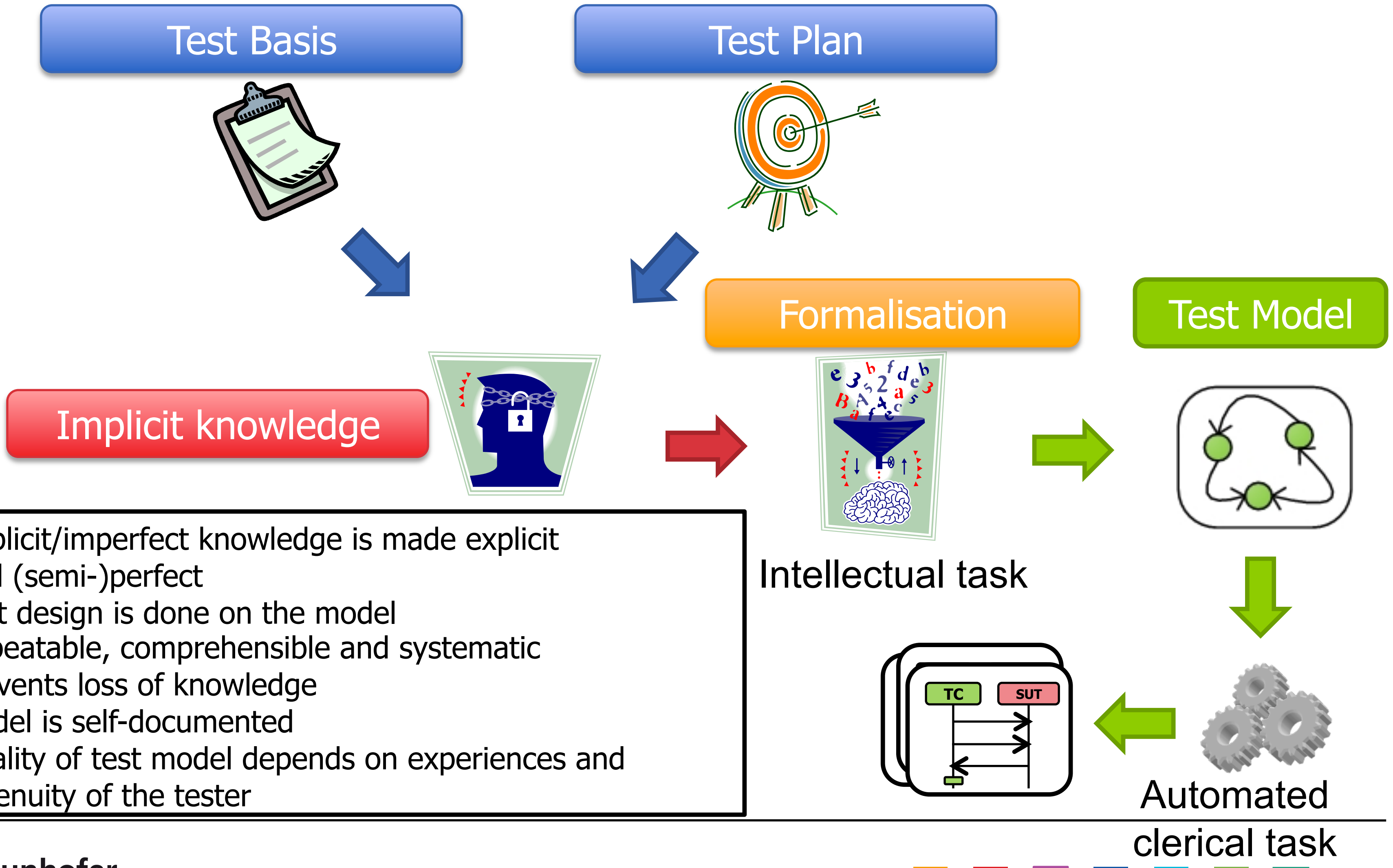
Test specification

Clerical task

- Implicit knowledge (mental model) of the test basis and system under test (SUT)
- Quality of test specification depends on the ingenuity and experiences of the tester
- Time consuming and prone to errors
- Not repeatability, lack of systematics
- Often not documented
- Loss of knowledge possible

Introduction

State of the art in automated test design – Model-Based Testing



Test Basis

Test Plan

Implicit knowledge

Formalisation

Test Model

Intellectual task

Automated clerical task

- Implicit/imperfect knowledge is made explicit and (semi-)perfect
- Test design is done on the model
- Repeatable, comprehensible and systematic
- Prevents loss of knowledge
- Model is self-documented
- Quality of test model depends on experiences and ingenuity of the tester

Introduction

State of the art in automated test design – Model-Based Testing (2)

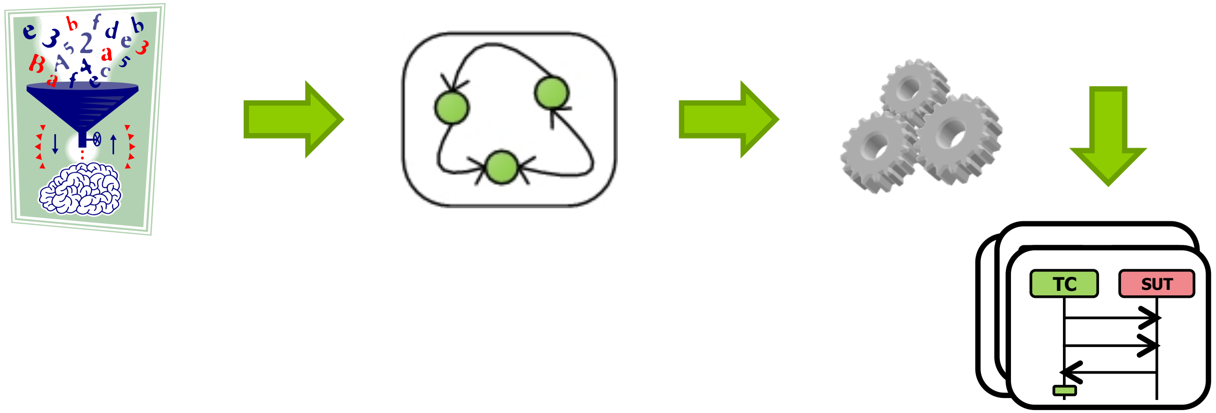
Clerical task

Intellectual task

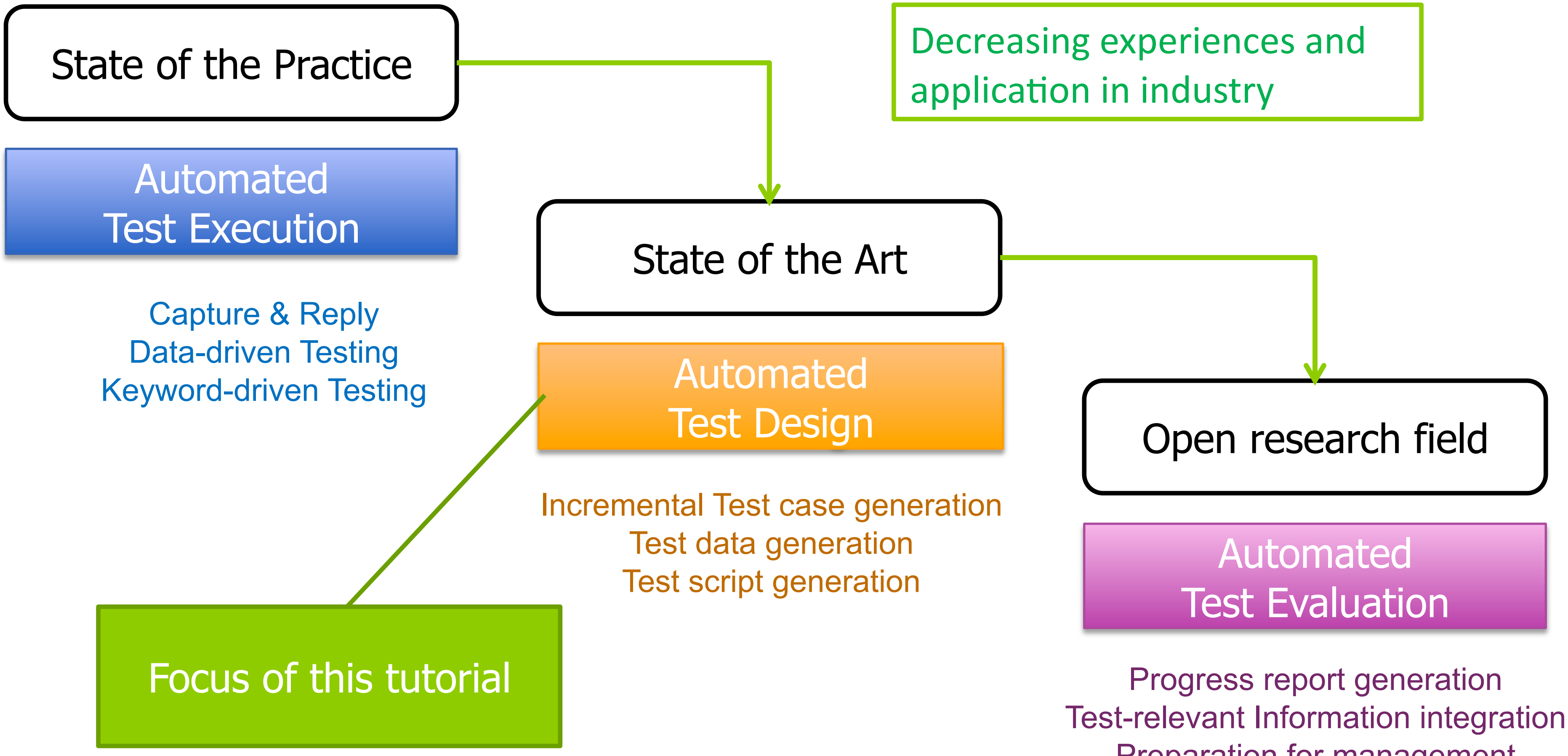
Automated Clerical task

Harvest

Model-based Testing



Introduction
Test automation in Industry



State of the Practice

Automated
Test Execution

Capture & Reply
Data-driven Testing
Keyword-driven Testing

State of the Art

Automated
Test Design

Incremental Test case generation
Test data generation
Test script generation

Decreasing experiences and application in industry

Open research field

Automated
Test Evaluation

Progress report generation
Test-relevant Information integration
Preparation for management
Automated metric calculation
Preparation of next cycle

Focus of this tutorial

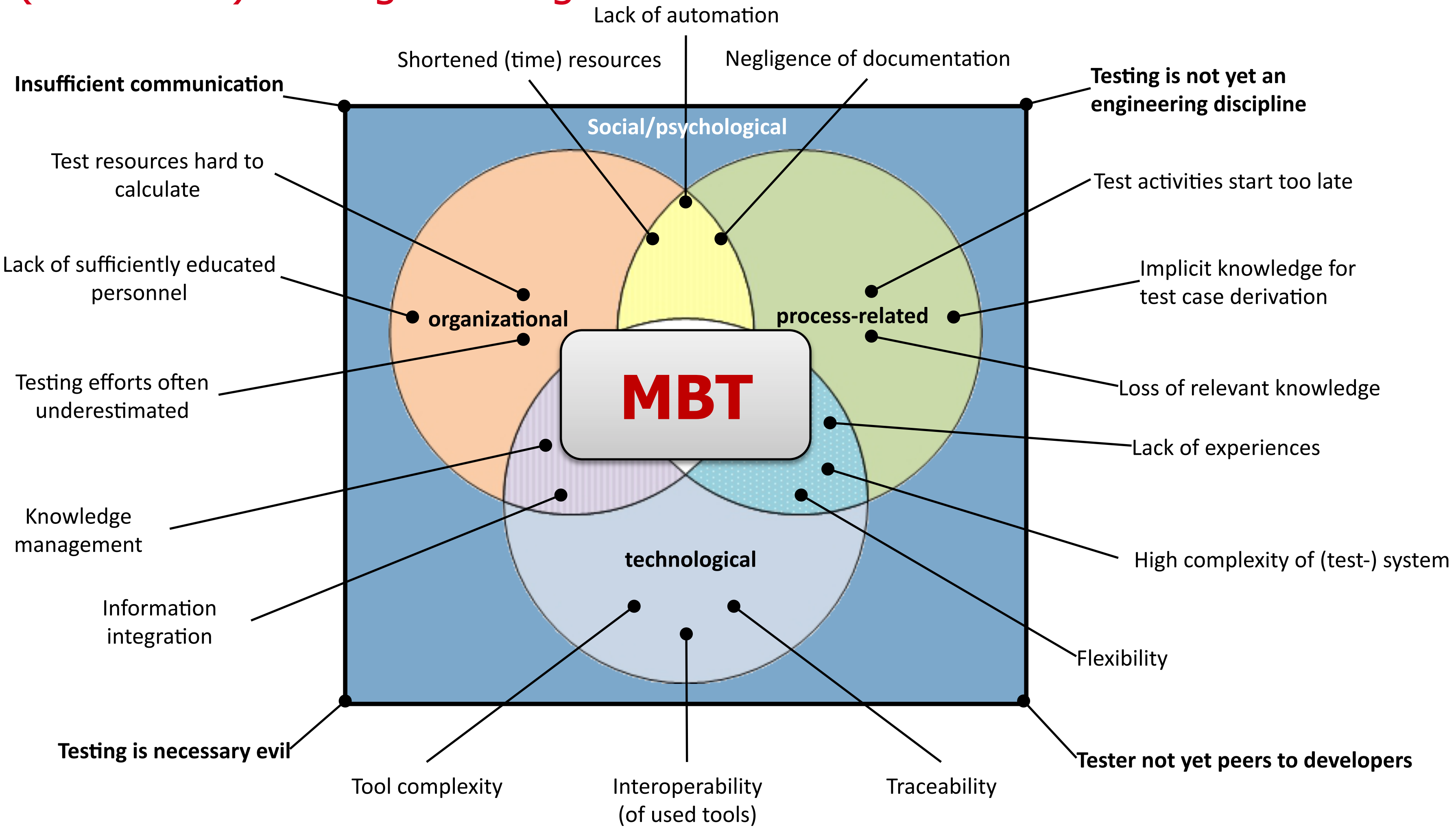
Agenda

- Introduction
- **Test automation with models**
- Industrial standards and notations
- Findings from industry
- Conclusion and discussion

What`s wrong with testing?

Test automation with models

(Traditional) Testing Challenges


Lack of automation
Shortened (time) resources
Negligence of documentation
Insufficient communication
Testing is not yet an engineering discipline
Social/psychological
Test resources hard to calculate
Test activities start too late
Lack of sufficiently educated personnel
Implicit knowledge for test case derivation
organizational
process-related
MBT
Testing efforts often underestimated
Loss of relevant knowledge
Lack of experiences
Knowledge management
High complexity of (test-) system
technological
Information integration
Flexibility
Testing is necessary evil
Tester not yet peers to developers
Tool complexity
Interoperability (of used tools)
Traceability

Test automation with models

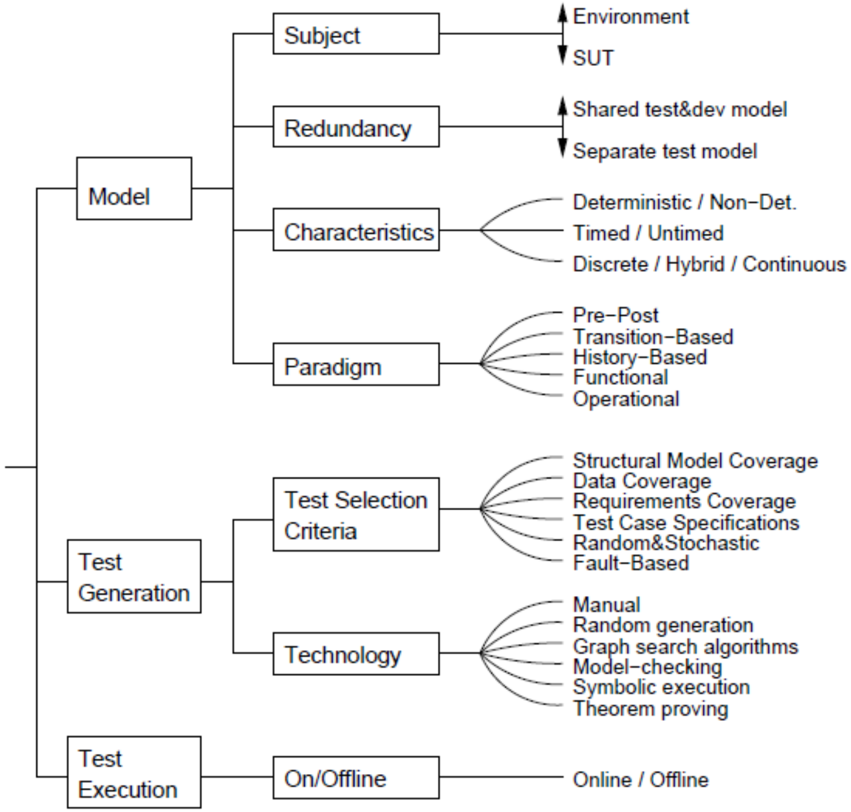
Definitions of Model-Based Testing

- Definition [EES11]
  "Model-based testing is an umbrella of approaches that generate tests from models."

- Definition [UTP]
  An umbrella of techniques that use (semi-)formal models as engineering artifacts in order to specify and/or generate test-relevant artifacts,
  such as test cases, test scripts, reports etc.
  (changed from [ES11]).

- MBT Taxonomy [Utt06]

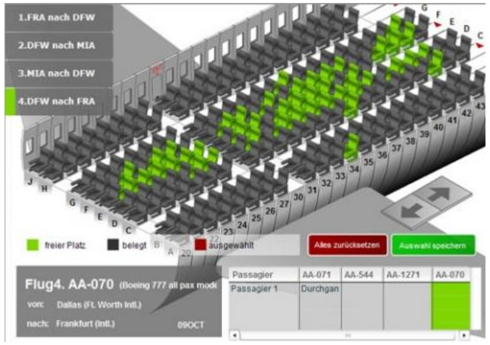
- Other taxonomies available!



Test automation with models
Classification of Models – General Definition

- Following Stachowiak‘s definition, a model is
  - *A view* on a real world concepts (maybe another models),
  - *An abbreviation* of the thing it represents by omitting irrelavant details for a given context, and
  - *Pragmatic* in the sense of being appropriate for the given context.

- Dörner added that models must possess
  - *Validity,* otherwise they would not represent the correct illustration and would not be pragmatic

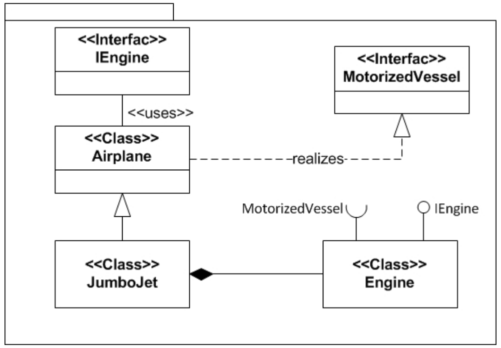


Flug4. AA-070 (Boeing 777 all pax mode

von: Dallas (Ft. Worth Intl.)

nach: Frankfurt (Intl.) 09OCT

Passagier	AA-071	AA-544	AA-1271	AA-070
Passagier 1	Durchgan			🟩



Test automation with models
Classification of Models – Technical Definitions

- Anneke Kleppe [Kle03]:

  "*A model is a description* ***(part of) a system*** *written in a* ***well-defined language****. A well-defined language is a language with* ***well-defined form (syntax)*** *and* ***meaning (semantics****), which is suitable for* ***automated interpretation*** *by a computer.*"

- UML Superstructure [UMLs11]:

  "*A model captures* ***a view*** *of a physical system. It is an* ***abstraction*** *of the physical system, with a* ***certain purpose****. This purpose determines what is* ***included*** *in the model and what is* ***relevant****. Thus the model completely describes those* ***aspects of the physical system*** *that are* ***relevant*** *to the* ***purpose*** *of the model, at the appropriate level of detail.*"

- MDA Guide [OMG03]

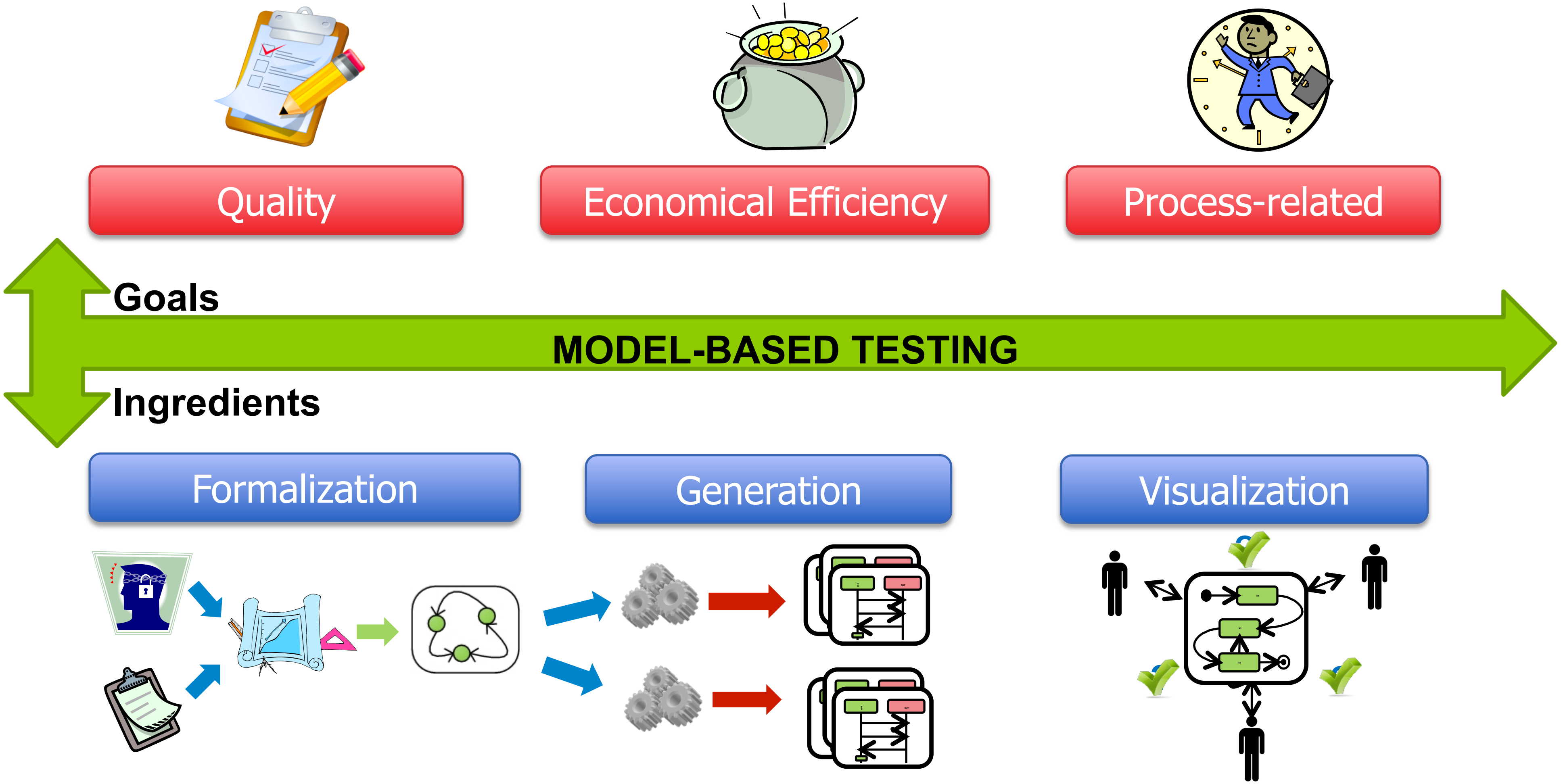
  "*A* ***formal specification*** *of the function, structure and/or behavior of an* ***application*** *or* ***system****.*"

- Chris Raistrick [Rai04]:

  "*A* ***formal representation*** *of the function, behavior, and structure of the system we are considering, expressed in an* ***unambiguous language***"

Test automation with models

Goals of Model-Based Testing – General Overview



Quality

Economical Efficiency

Process-related

Goals

MODEL-BASED TESTING

Ingredients

Formalization

Generation

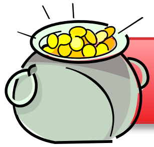
Visualization

Test automation with models

Summary: Most Significant Impacts of Model-Based Testing

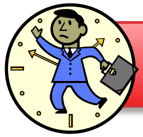
Quality

- Increased traceability
- Tightly integrated information in test model
- Higher quality of relevant specifications
- Automated quality control of test artifacts
- Improved, self-contained documentation
- Complexity control by abstraction
- Improved documentation
- Prevents loss of knowledge



Economical Efficiency

- Lower time-to-market
- Increased productivity: Faster design of test cases
- Increased productivity through automation
  - Reuse of existing test artifacts
  - Higher portability
  - Higher maintainability
- Automated coverage analysis and other statistical analysis
- Lower test design and execution costs
- Improved resources management



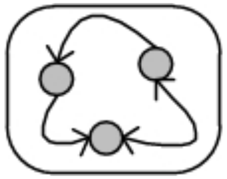
Process-related

- Early validation of requirements
- Early validation of system specification
- Prioritization of test cases facilitates test management
- Early specification of test cases
- Automated test (re-)generation
- Automated generation of reports and analysis
- Increased opportunities for cost-reduction through outsourcing
- Visualization leads to higher understandability
- Improved communication between stake holders

Test automation with models

System, Test and Additional Models

System Model



- An internal view of the system, its components, interfaces and data types
- Describes how a system is constructed
- Specifies a system’s design (design model)
- Constitutes the system specification an actual implementation must comply with

Test Model

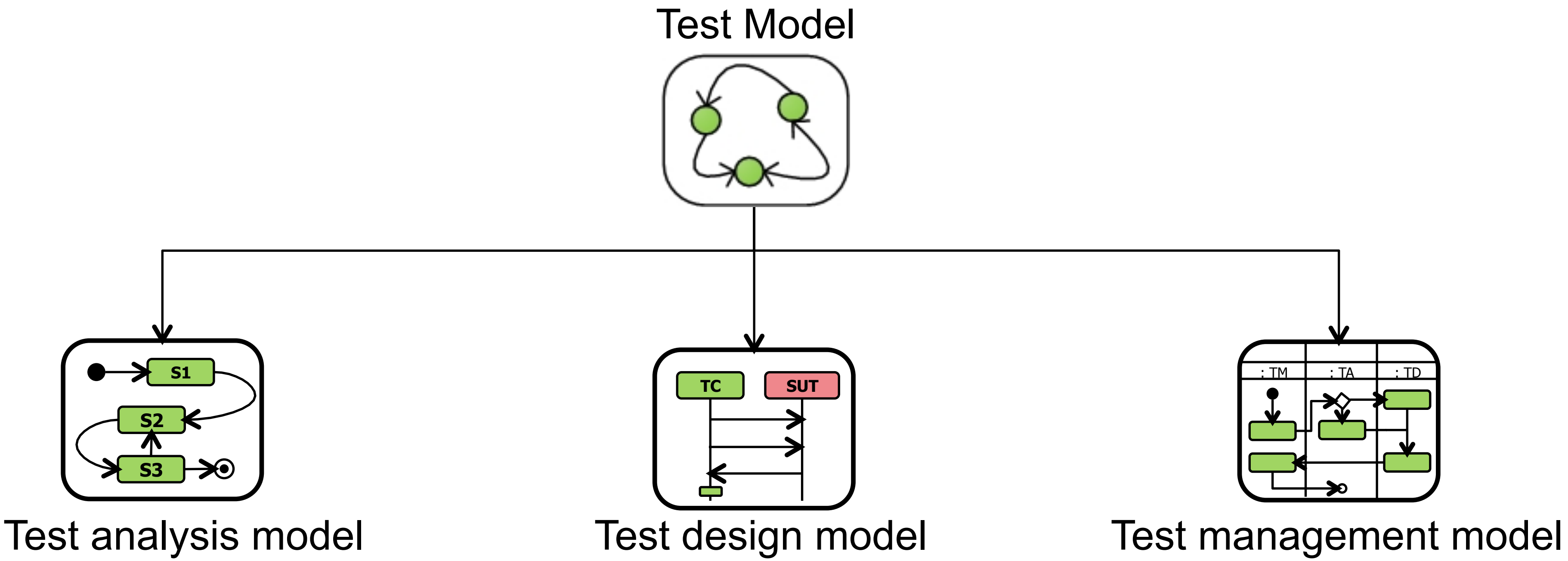
- Describes how a system is to be used/tested
- Neglects internal aspects, emphasizes the externally observable behavior of
- May be used for test case generation
- May reuse artifacts of the system and/or from additional models

Additional Models

- A view on additional aspects related to the system
- Describes information beyond system or test models
- E.g. Requirements models, operational usage models, risk models, work flow models, environment models

Test automation with models

Views on Test Models



Test Model

Test analysis model

Test design model

Test management model

- Intended behavior of the system under test
- Used for test case generation

- Test data
- Test descriptions
- Test cases
- Test suite

- Test strategy
- Test plan
- Test requirements
- Test directives
- Test results

Test automation with models
Abstractions in Model-Based Testing

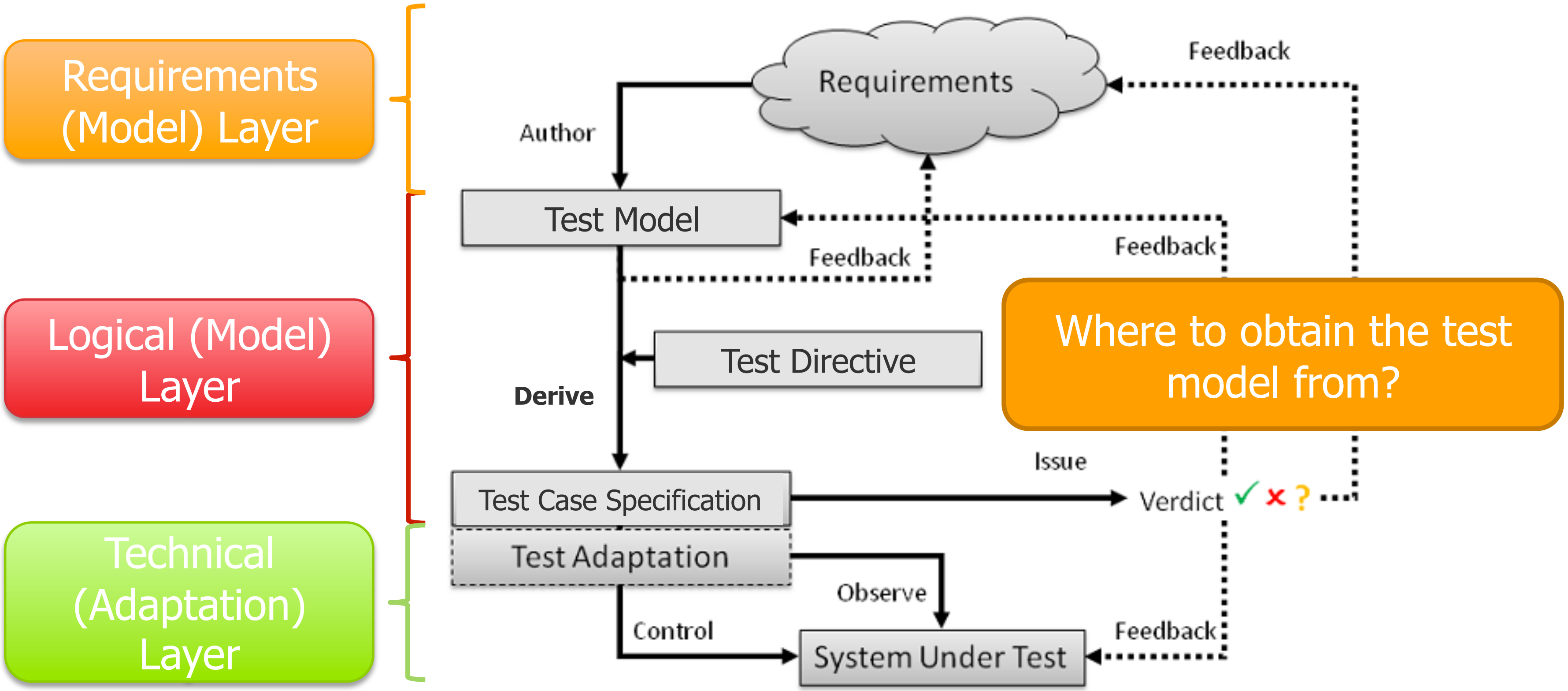
- Functional abstraction
  - Concentrate on aspects of the system pertinent to the target of the test level
  - Divide functional to be tested for better maintenance
- Data ab
  - Abs
  - Log
- Commu
  - The actual communication with the SUT might be too complex
  - Single operation call in the model is realized to several calls in the adapter
- Temporal abstraction
  - Abstraction from physical timer, time units or granularities

Abstraction leads to simpler test models compared to the actual system or ist specificaction.

Complexity needs to be faced during test realization

Source: [Pre]

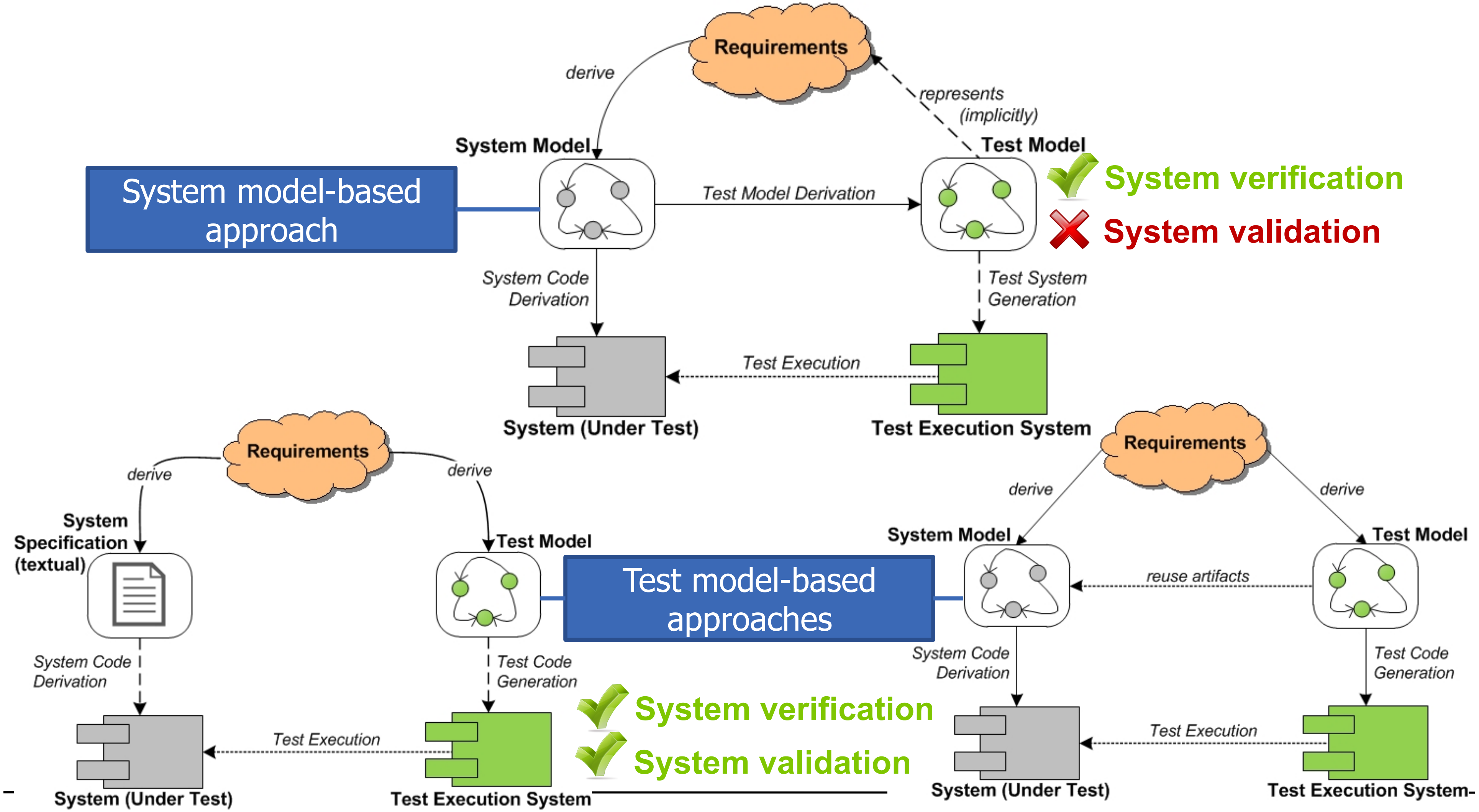
Test automation with models
Abstraction Levels of Test Models



Where to obtain the test model from?

Source: [EES11]

Test automation with models
Approaches to Model-Based Testing



System model-based approach

✓ System verification

✗ System validation

Test model-based approaches

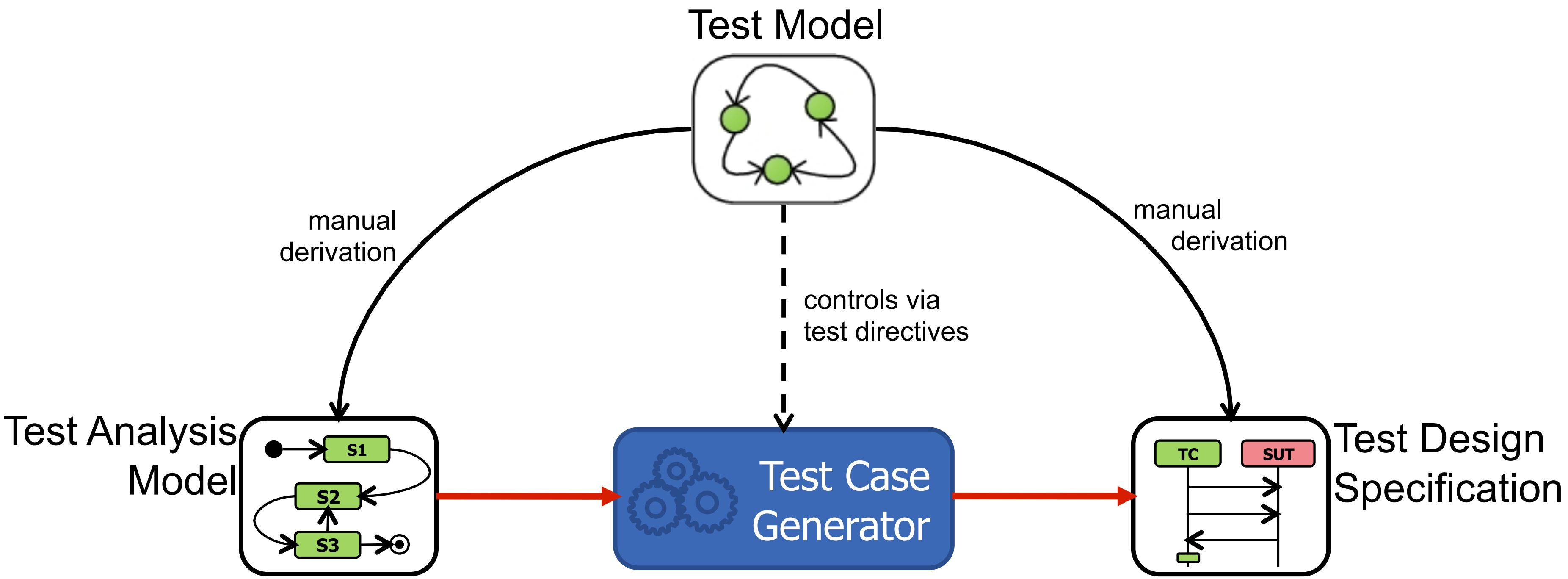
✓ System verification

✓ System validation

Source: [Schief]

Test automation with models

Automated and manual test design



The system‘s intended behavior from a tester‘s perspective. Serves as foundation for test case generation.

Instructions how to derive test cases from a behavioral description. The test plan determines what coverage or generation criteria to apply to the test case generator

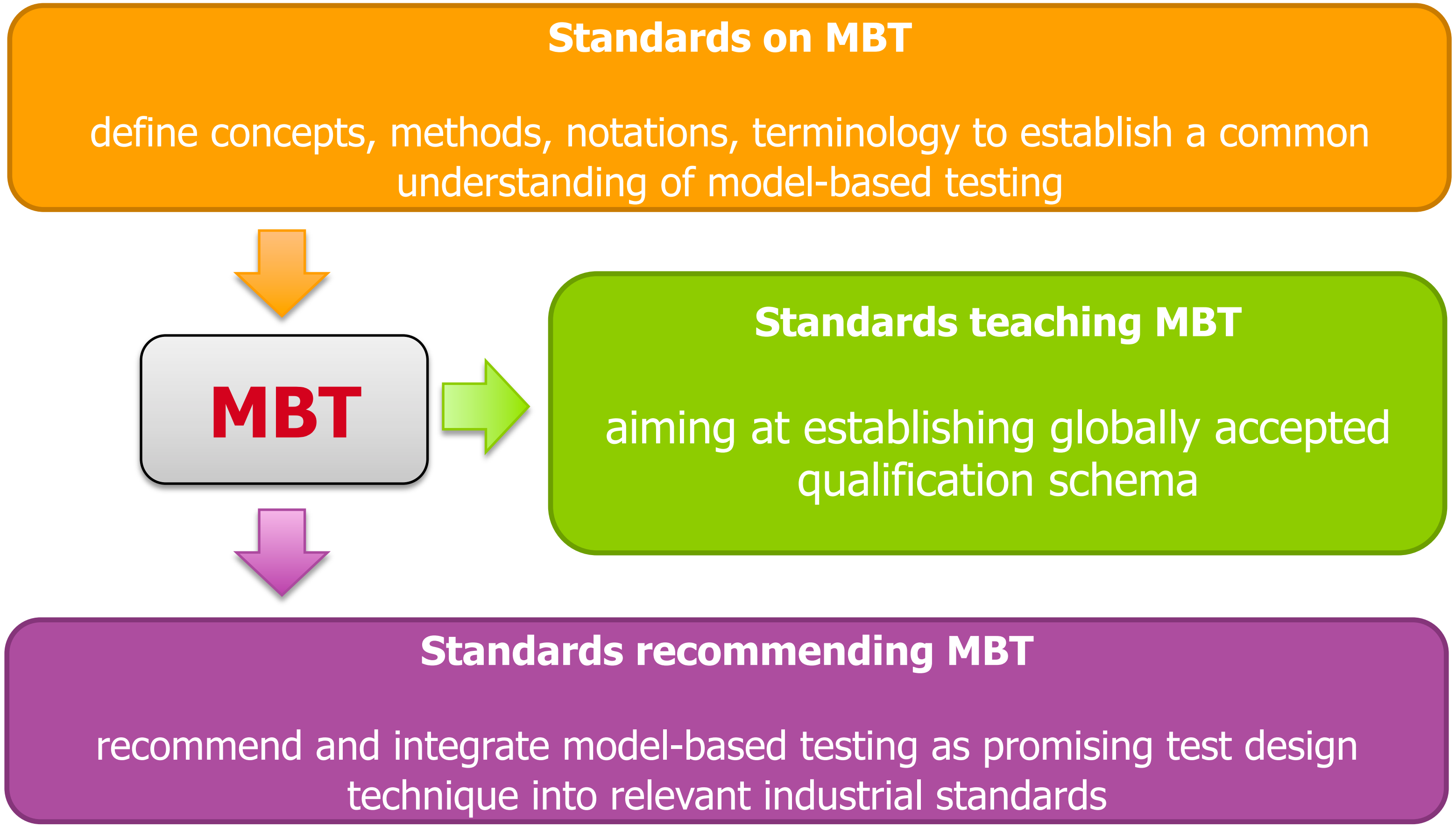
Specification of concrete test cases how to test the system. Developed accordingly to test objectives.

Agenda

- Introduction
- Test automation at a glance
- Models for test automation
- **Industrial standards and notations**
- Findings from industry
- Conclusion and discussion

Management View of MBT

Standardization Efforts on Model-Based Testing



Standards on MBT

define concepts, methods, notations, terminology to establish a common understanding of model-based testing

MBT

Standards teaching MBT

aiming at establishing globally accepted qualification schema

Standards recommending MBT

recommend and integrate model-based testing as promising test design technique into relevant industrial standards

Industrial Standards and Notations
Standards on Model-Based Testing

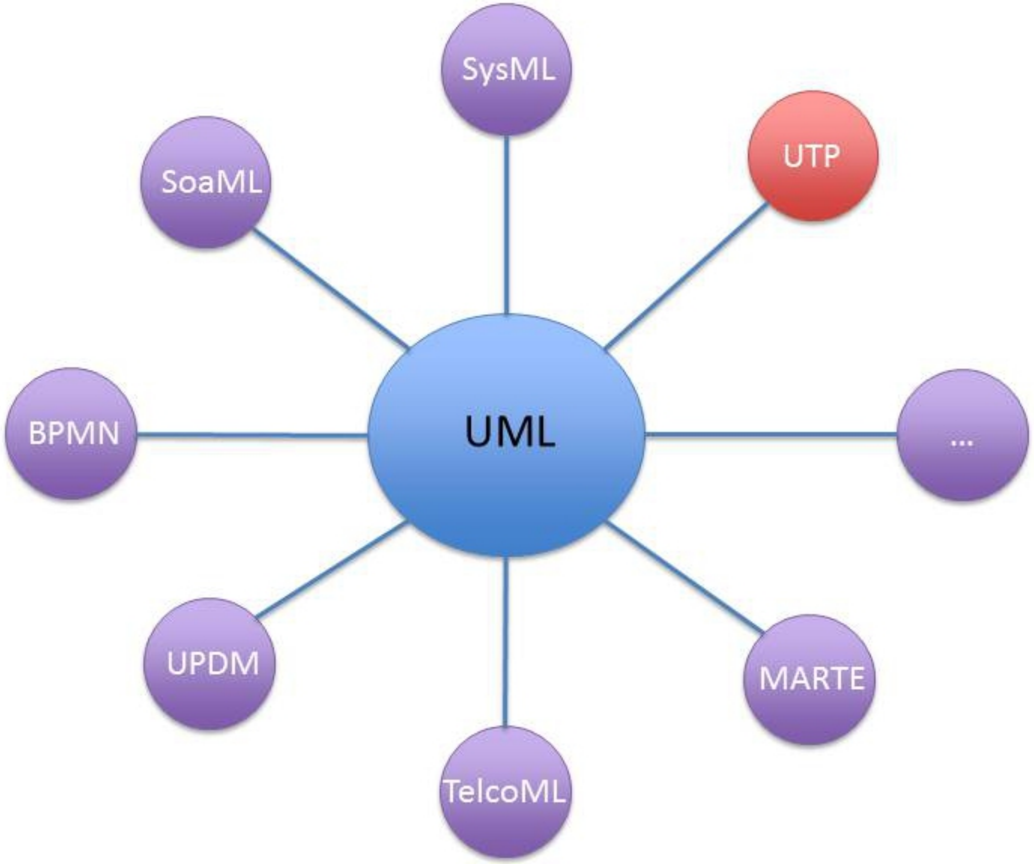
- OMG
  - UML Testing Profile (UTP), Version 1.2
  - Test Interchange Format (TestIF), Version 1.0 Beta 1

- ETSI
  - TR 102 840 V1.2.1 (2011-02):
    Methods for Testing and Specifications (MTS);
    Model-based testing in standardisation
  - ES 202 951 V1.1.1 (2011-07):
    Methods for Testing and Specification (MTS); Model-Based Testing (MBT);
    Requirements for Modeling Notations
  - Test Description Language (TDL) – *under construction*

- IEEE
  - 1671: Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML

Standards on Model-Based Testing

UML Testing Profile in the UML Ecosystem



Standards on Model-Based Testing
Goals of UML Testing Profile

- UML natively lacks concepts for testing of systems/software

- A profile based upon UML, which
  - enables the definition and/or generation of model-based test specifications, including structural and behavioral aspects of the system under test (SUT) using UML, and
  - bridges the gap between engineers (e.g. system and test engineers)

- Provide a concrete standardized notation that enables user to conduct testing in a model-based way (fulfills all ETSI’s requirements for model-based testing appropriate notations)

- Reuse of or combination with other horizontal domain-specific profiles of the OMG, e.g. MARTE, SysML, SoaML, …

Standards on Model-Based Testing

What is UML Testing Profile made for?

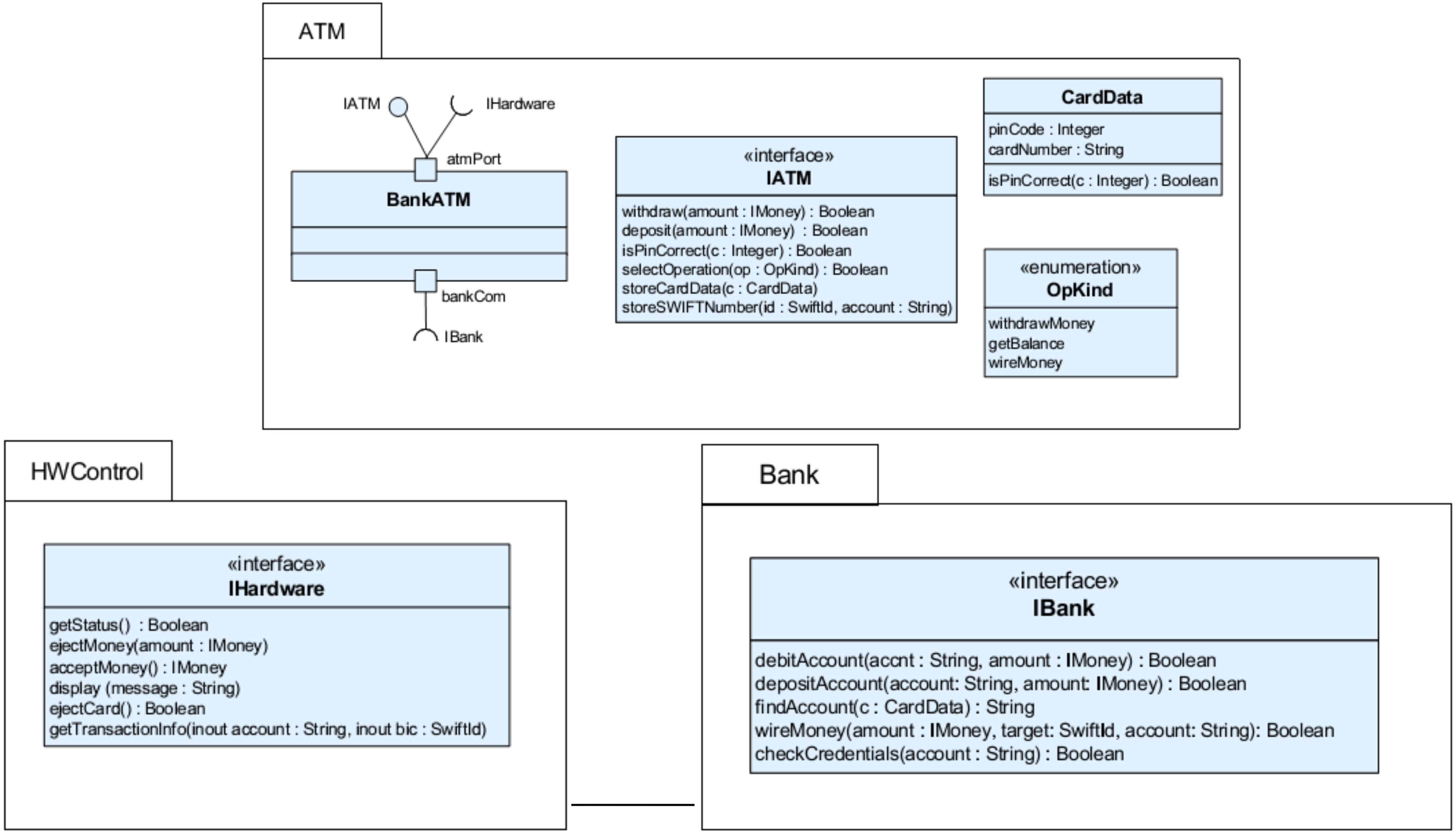
- Domain-independent test modeling for dynamic testing approaches
  - Test environments
  - Test configurations
  - Test case specifications (including test case derivation)
  - Test data specifications/values
- Provides means for both white box and black box testing approaches
- Managing and visualization of test results
- Documentation of the test process (e.g. report generation)
- Integration of best practices such as keyword-driven testing, equivalence class testing, etc.
- Combination with other profiles (e.g. SysML, MARTE, SoaML)
  - E.g. to achieve requirements traceability, …

Standards on Model-Based Testing
… and what is out of scope?

- Test methodology
- Modeling of test processes
- Some static test approaches such as *audits* and *reviews*
- Test case generation directives (i.e. how to carry out the test case generation process en detail)
- Test data generation directives (i.e. how to carry out the test data generation process en detail)
- Some kinds of integration testing

Standards on Model-Based Testing

UML Testing Profile – An Example (1)



ATM

IATM

IHardware

atmPort

BankATM

bankCom

IBank

«interface»
IATM

withdraw(amount : IMoney) : Boolean
deposit(amount : IMoney) : Boolean
isPinCorrect(c : Integer) : Boolean
selectOperation(op : OpKind) : Boolean
storeCardData(c : CardData)
storeSWIFTNumber(id : SwiftId, account : String)

CardData

pinCode : Integer
cardNumber : String

isPinCorrect(c : Integer) : Boolean

«enumeration»
OpKind

withdrawMoney
getBalance
wireMoney

HWControl

«interface»
IHardware

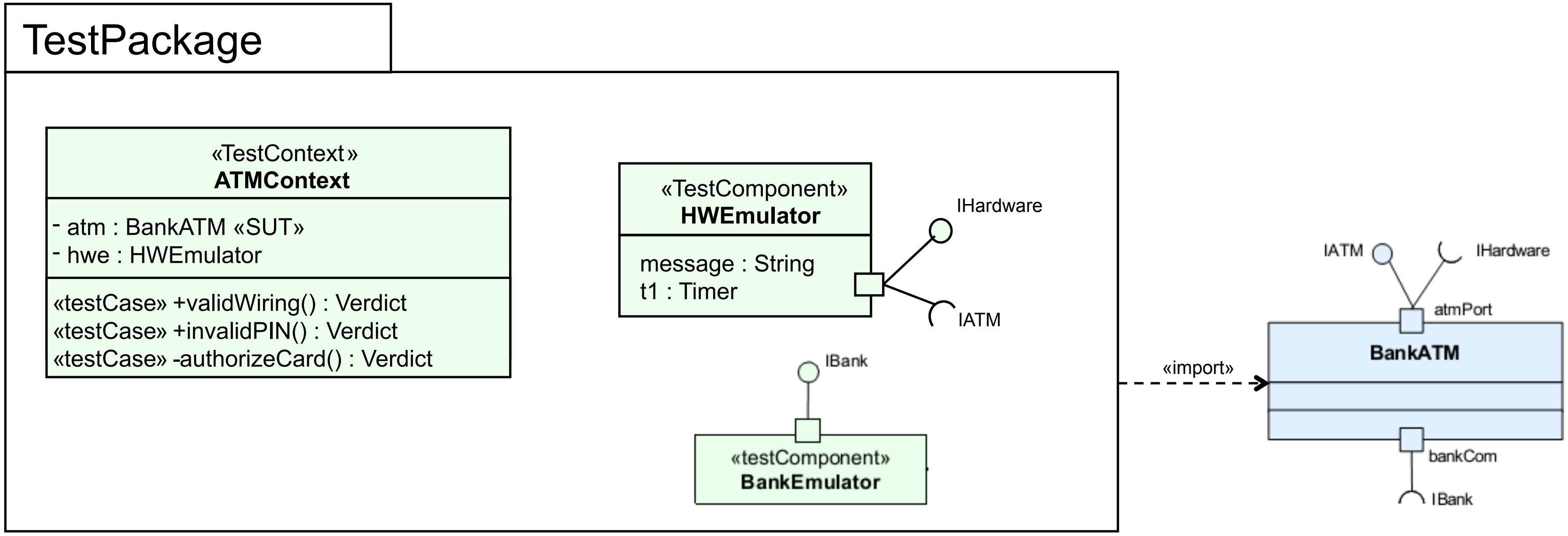
getStatus() : Boolean
ejectMoney(amount : IMoney)
acceptMoney() : IMoney
display (message : String)
ejectCard() : Boolean
getTransactionInfo(inout account : String, inout bic : SwiftId)

Bank

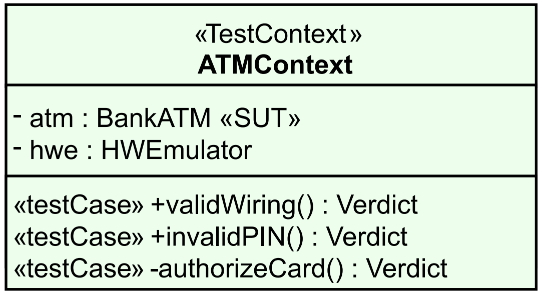
«interface»
IBank

debitAccount(accnt : String, amount : IMoney) : Boolean
depositAccount(account: String, amount: IMoney) : Boolean
findAccount(c : CardData) : String
wireMoney(amount : IMoney, target: SwiftId, account: String): Boolean
checkCredentials(account : String) : Boolean

Standards on Model-Based Testing
UML Testing Profile – An Example (2)



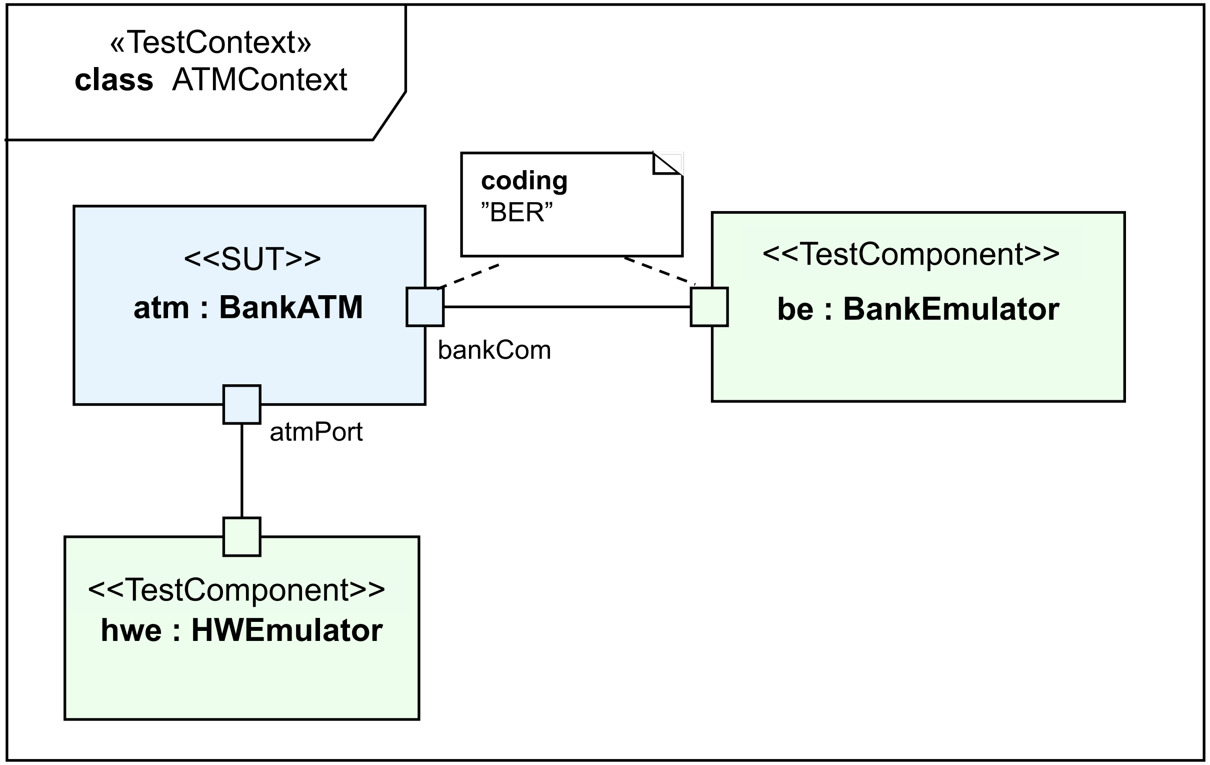
Standards on Model-Based Testing
UML Testing Profile – An Example (3)



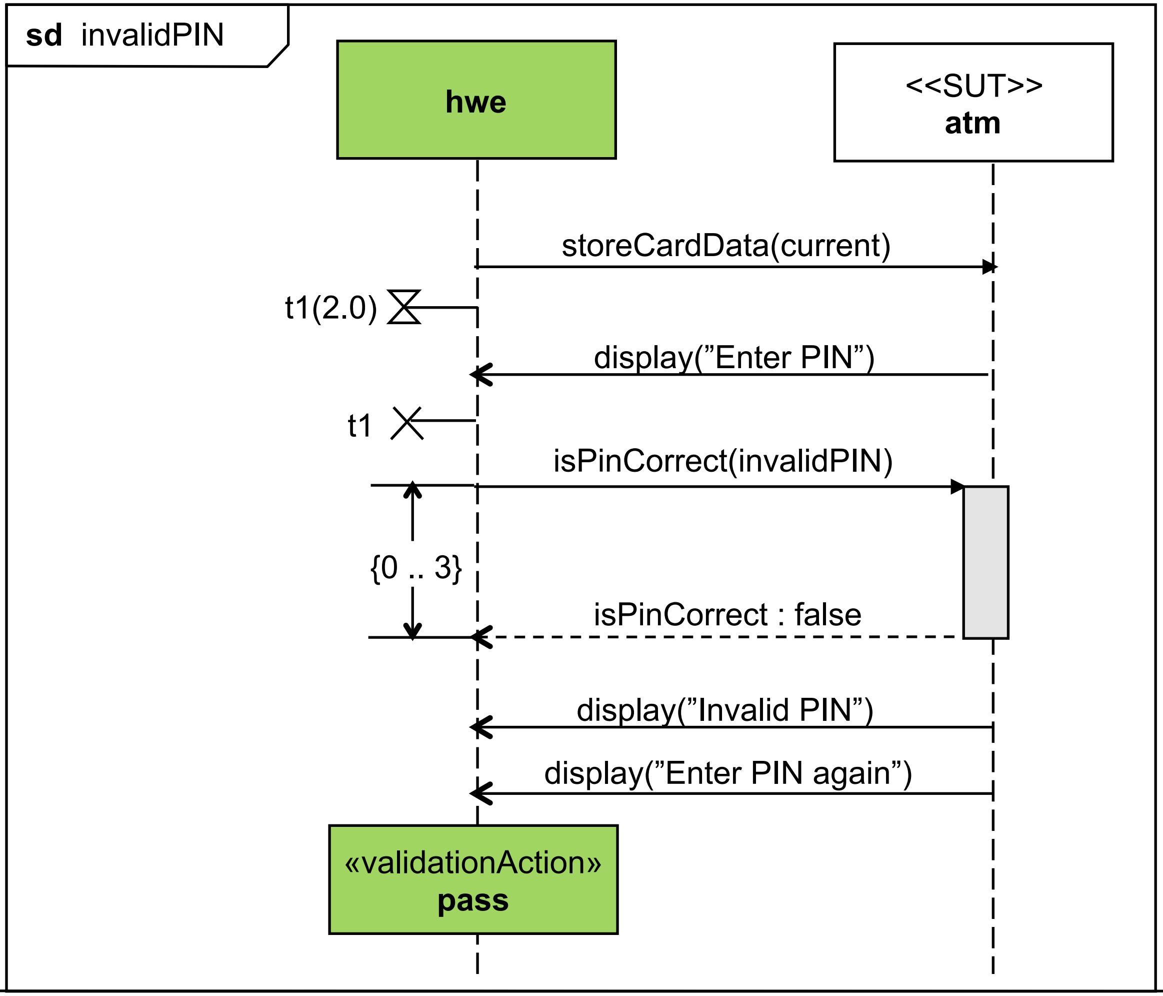
«TestContext»
ATMContext

- atm : BankATM «SUT»
- hwe : HWEmulator

«testCase» +validWiring() : Verdict
«testCase» +invalidPIN() : Verdict
«testCase» -authorizeCard() : Verdict



Standards on Model-Based Testing
UML Testing Profile – An Example (4)



Standards on Model-Based Testing
Perception by Industry

- UTP was/is not widespreadly used in industry
  - Lack of experiences with UML 2
  - Insufficient support of mature UML 2 tools
  - Model-based testing was/is rather academic „vodoo"
  - Lack o
- Criticisms of
  - Insuffi
  - Missing
  - Inadequate readability of the specification document
  - MOF-based metamodel and native UML profile was confusing

UTP was ahead of its time

Standards on Model-Based Testing
RFI for UML Testing Profile v2.0

- There will be no UTP 1.3!
- A new RFI was issued on 13th of September, 2012 (Wednesday)
- General question categories (45 questions alltogether)
  - Information about responder
  - MBT in general
  - UTP v1 Feedback
  - Support or test modeling
  - Tools and Techniques
  - Questions for tool vendors
  - Correlation with other standards
  - Optional: Concrete questions regarding UTP and existing OMG standards
- Responses will be discussed at the forthcoming OMG technical meeting in June!
  - Expected to submit an RFP in 2013

Industrial Standards and Notations

Standards Recommending Model-Based Testing

- Model-based testing slowly gets into quality standards
- Two recently renewed/incepted standards recommend model-based testing for particular Safety Integrity Levels (SIL)

Standard	Release Date	Technique	(A)SIL 1	(A)SIL 2	(A)SIL 3	(A)SIL 4
ISO/IEC 61508	2010	Model-based Testing	+	+	++	++
ISO 26262 - 4	2011	Back-to-Back Test*	+	+	++	++

Note from ISO 26262-4:
A back-to-back tests compares the responses of the test objective with the responses of the simulation model to the same stimuli, to detect differences between the behavior of the model and its implementation.

[Weiss]

Industrial Standards and Notations
Standards recommending MBT - ISO 29119

All testing uses the concept of a model representing the test item's expected behaviour being available as the test basis… Traditionally, the tester uses the model to manually derive test inputs and expected results

Model-based testing uses a fundamentally different approach, but still based on a model of the expected behaviour.

The difference is that with model-based testing the model has to be formal enough and detailed enough so that an automated tool can analyse the model to create complete test cases (test inputs and expected results – the model will act as the test oracle)

A further requirement for model-based testing is that the automated test cases can be automatically executed on the test item and the actual results compared with the expected results.

The use of a model-based testing approach should therefore be considered where the risk of application failure is high and the risk of future maintenance costs is high.

Industrial Standards and Notations

Standards Teaching Model-Based Testing – Certified Model-Based Tester

- Motivation for and basics of MBT
  - Brief repetition of testing and test process basics
  - Learn about possible improvement goals
  - Benefits of MBT
  - Limitations of MBT
- C

Adopted by iSQI

First classes have been taught and certifications have been made

Plan to submit this schema to ISTQB in the near future

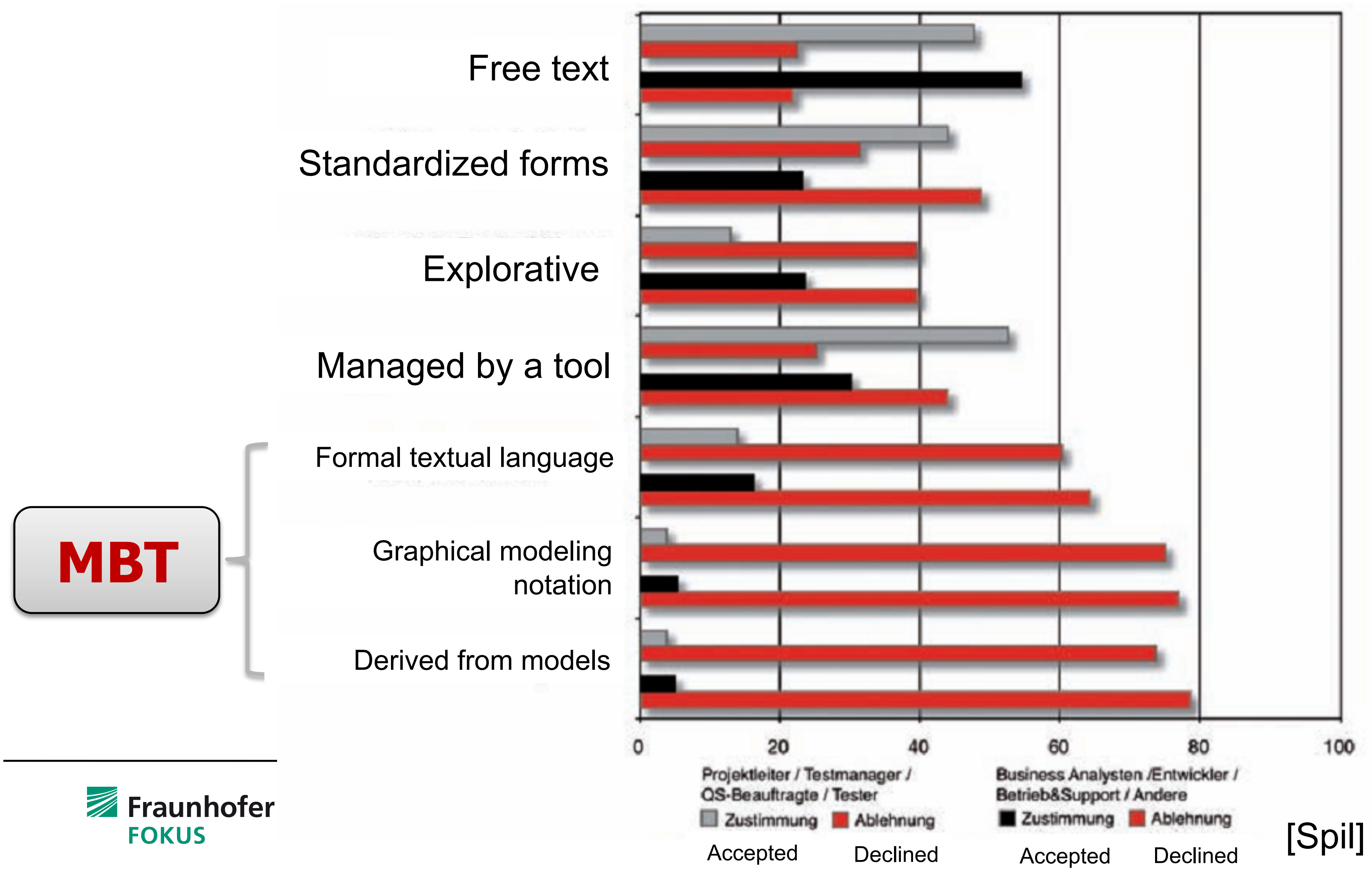
- Development of (test) models
  - General concepts for modeling in software engineering
  - How to model test models (notational)
- Test case generation and test generation strategies
- ROI considerations

Agenda

- Introduction
- Test automation at a glance
- Models for test automation
- Industrial standards and notations
- **Findings from industry**
  - **Is model-based testing already accepted by industry?**
  - Challenges and recommendations for MBT integration
  - ROI considerations and improvement potential
  - Selected tools for industrial application
- Conclusion and discussion

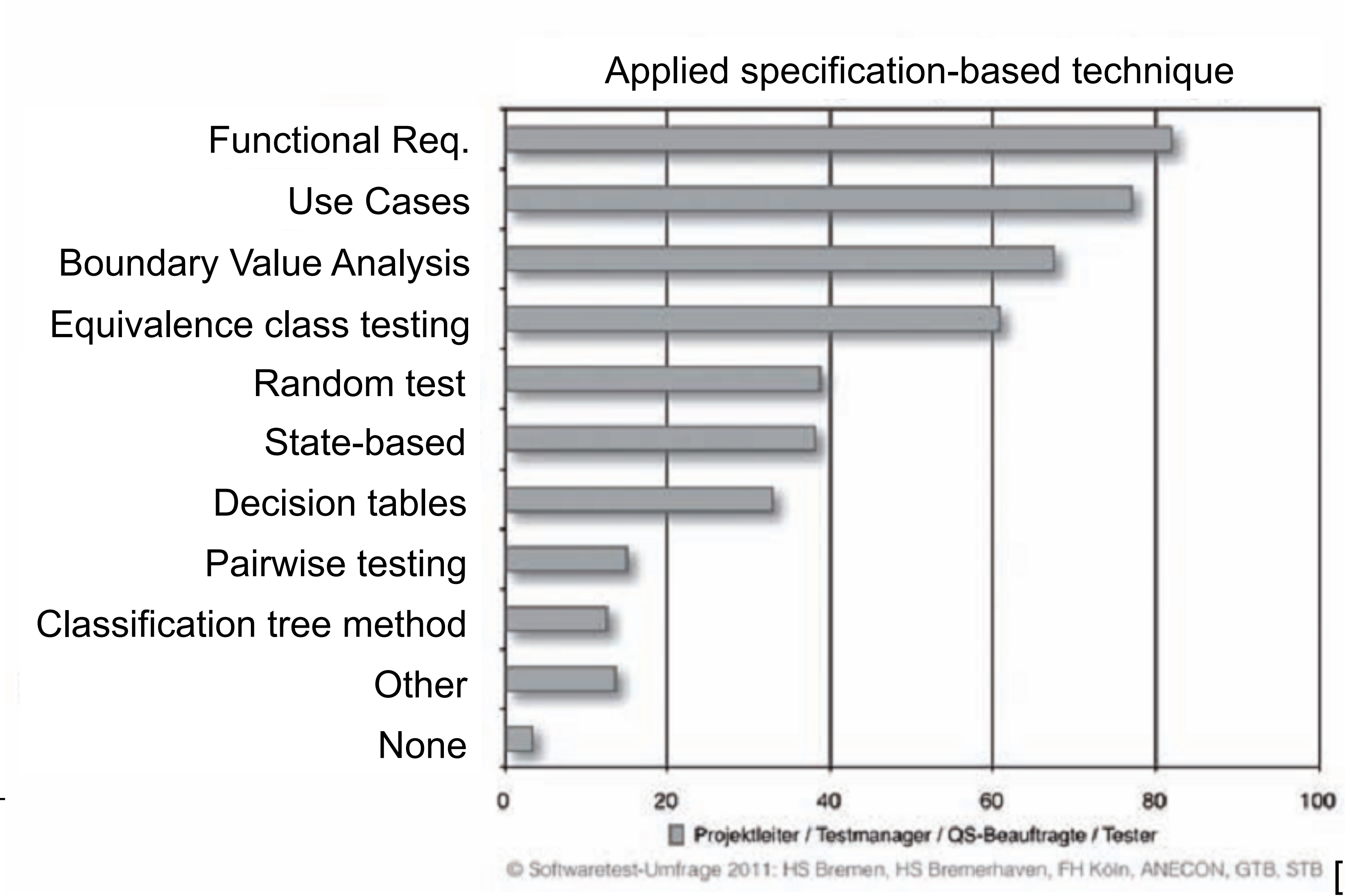
Findings from Industry

Has Model-Based Testing reached Industry acceptance?



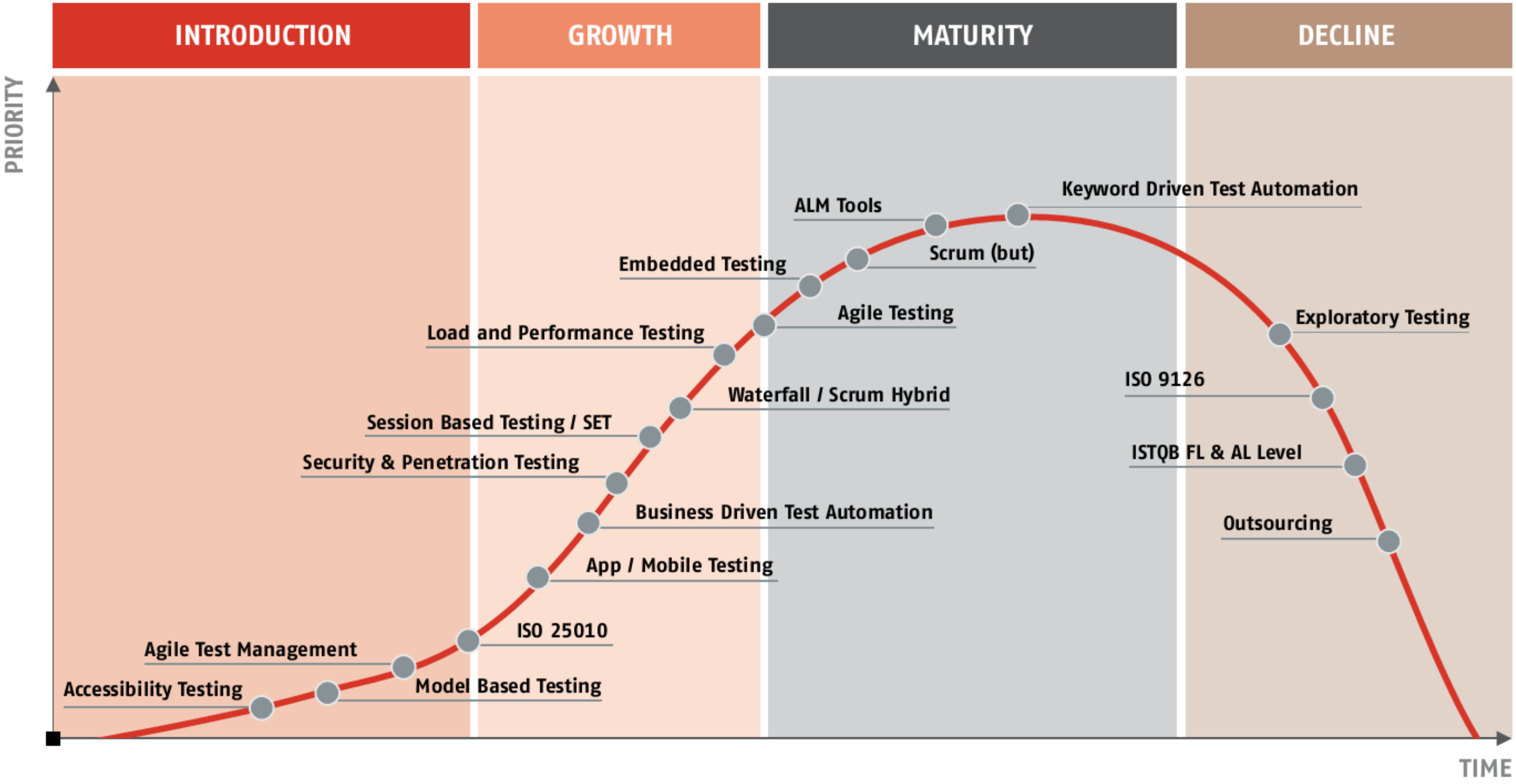
[Spil]

Findings from Industry
Has Model-Based Testing reached Industry acceptance? (2)



© Softwaretest-Umfrage 2011: HS Bremen, HS Bremerhaven, FH Köln, ANECON, GTB, STB [Spil]

Findings from Industry
Has Model-Based Testing reached Industry acceptance? (3)



[SwissQ]

Fraunhofer FOKUS

Agenda

- Introduction
- Test automation at a glance
- Models for test automation
- Industrial standards and notations
- **Findings from industry**
  - Is model-based testing already accepted by industry?
  - **Challenges and recommendations for MBT integration**
  - ROI considerations and improvement potential
  - Selected tools for industrial application
- Conclusion and discussion

Findings from Industry
Organizational Challenges of Model-Based Testing

- Unrealistic expectations: MBT is no silver bullet for all testing problems
- Lack of modeling culture and education
- Inappropriate process
- Process m
  - Brea
  - Intro
- Educationa
  - How to educate testers according to the necessary skills for MBT?
- Quality control
  - Modeling guidelines and associated model checking routines
- Establish integrated and automated tooling landscape for MBT

Organizational costs considerations

Process migration

Tooling

Education

Findings from Industry
Cost Considerations on Model-Based Testing

- MBT tool costs: the costs of acquiring new tools and frameworks in order to implement the MBT approaches in a broader way.
- MDE tool costs: the implementation of MBT can be coupled with the implementation of model-driven engineering processes. To fully exploit the advantages of MBT, also an MDE infrastructure (tools, methodology) is recommended.
- Adaption costs to the company's tool and process infrastructure: the MBT methodology and tool platform need to be fine-tuned with respect to the company's development processes, best practices, and domain requirements. Moreover, a fine-tuning for particular projects or at least project categories is often needed.
- Qualification costs: the implementation, maintenance, and integration of MBT procedures require a higher level of expertise than traditional test activities. The costs for qualification and training as well as for new experts have to be considered.
- Roll-out costs when changing existing methods, procedures, and best practices.

Findings from Industry
Technical Challenges on Model-Based Testing: Tooling

- Task of integrating a new tool into an existing process/tool landscape should not be underestimated.
- Tool needs to be tailored to the modeling and testing methodology
  - Wizards, patterns, templates
- Collaborative work on models
  - Changes tracking, model diff and merge
  - Semantic consistency check
  - Design-/Architecture consistency check
- Validation of test models
  - Syntax checking is not enough: semantic consistency also needs to be assessed
- Maintainability of test models
  - Model size grows rapidly
  - Treat models as assets
- Means for Simulation & Verification
  - Rapid prototyping

Findings from Industry
Technical Challenges on Model-Based Testing: Modeling

- Creation of models for testing is not trivial
  - What language and notation is appropriate for the given system
  - What kind of behavior shall be used
  - Size of behavioral descriptions for test case generation

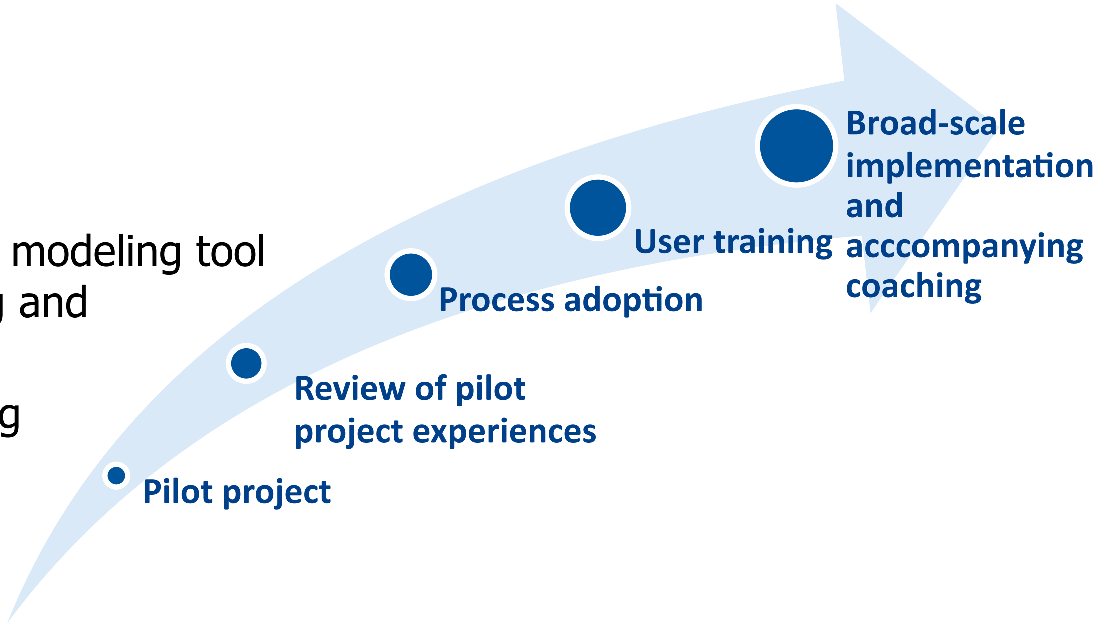
- Reuse of existing test model artifacts
  - Horizontal reuse: e.g. new test model artifacts from existing ones
  - Vertical reuse: e.g. new system test model artifacts from legacy integration test model artifacts

- Legacy artifacts
  - Reverse-engineering of existing artifacts (e.g. system data, architecture, behavior) for reuse.
    - Reuse of system data specifications (ASN.1, XML, IDL…)
    - Reuse of SUT architectures (SOAP, IDL,…)
    - Visualization and reuse of test behavior from test automation scripts

Findings from Industry
Migration towards Model-Based Testing

- Migration to MBT is similar to migration to other test automation approaches
- Migration to MBT encompasses four main phases:
  1. MBT process definition and integration with established processes
  2. Tool selection and training
  3. Piloting
  4. Deployment

- We recommend to choose
  - An already used/customized modeling tool extended with test modeling and test generation support
  - A testing team with modeling experiences
  - A pilot with manageable functionality and leveraged time constraints



Pilot project
Review of pilot project experiences
Process adoption
User training
Broad-scale implementation and acccompanying coaching

Agenda

- Introduction
- Test automation at a glance
- Models for test automation
- Industrial standards and notations
- **Findings from industry**
  - Is model-based testing already accepted by industry?
  - Challenges and recommendations for MBT integration
  - **ROI considerations and improvement potential**
  - Selected tools for industrial application
- Conclusion and discussion

Findings from Industry
ROI considerations and improvement potential

- 100% req coverage by 2/3 of manually created test cases with MBT
- 15% time improvement for initial creation of test assets
- 40% time improvement for each increment/ test cycle
- break-even during 2nd year after roll-out

[Szé11]

- 10x-20x savings in subsequent tested product iterations
- test creation time savings: 55% average
- 100% documentation generation
- SUT coverage increased by 30-50%
- Fault detection increased by 20%-40%
- Maintenance costs decreased by 50%-90%

[Kon11]

Effort per TC creation in incremental versions: ~74%

[Göt10]

Time savings: 14x compared to manual testing

[Kar11]

- 90% productivity improvement in case study 1
- 88% productivity improvement in case study 2

[Suh11]

17% time savings (including educational time for personnel) compared to manual test case derivation

[Far02]

Findings from Industry
ROI considerations and improvement potential (2)

Case Study/ Company	Tool	Effort (no MBT)	Effort (MBT)	Cost saving
Ericsson	Conformiq	20h/Test case	5.5h/Test Case	73%
Trapeze	Siemens	2.67h/Test case	0.67h/Test case	75%
sepp.med	MBTsuite	2.05h/test case	1.36h/Test Case	43%
Microsoft	SpecExplorer	2.37 days/ requirement	1.39 days/ requirement	42%
Forrester	Conformiq	6.396.565$	1.288.94$	30% initial 84% 2nd cycle

Source: [Weiss2]

Findings from Industry
ROI considerations and improvement potential (3)

Efforts in h	Traditional approach	MBT approach
Analysis of Test Basis	33	33
Modeling Test Analysis Model	-	40
Test Design	100	14
Clarification discussions	10	8
In total:	**143**	**95**

Source: [Weiss2]

Agenda

- Introduction
- Test automation at a glance
- Models for test automation
- Industrial standards and notations
- **Findings from industry**
  - Is model-based testing already accepted by industry?
  - Challenges and recommendations for MBT integration
  - ROI considerations and improvement potential
  - **Selected tools for industrial application**
- Conclusion and discussion

Findings from Industry
Tools for Model-Based Testing

Tool	URL	Target Domains	Test model	Test generation criteria	Test scripting
CertifyIT	http://www.smartesting.com	Software	BPMN or UML	Test data and verification points	Textual test plans
Conformiq Designer	http://www.conformiq.com/	Datacom and Telecom	UML-like State Machines	Requirements-driven test generation, black-box test design heuristics	Textual test plans and executable test cases in Java, etc.
Spec Explorer 2010	http://research.microsoft.com/en-us/projects/specexplorer/	Software	Spec#	Transition coverage	Executable test cases in C# or on-the-fly testing
Tedeso 3.0	http://www.imbus.de/english/imbus-testbench/modules/managed-model-based-testing/	Software	UML-like Use Case Activity Diagrams	Model and data coverage	Executable test cases in C++, etc.
TestCast Generator BETA	http://www.elvior.com/motes/generator	Telecom, transport, defense	UM-like State Machines	State, transition and decision coverage	Executable test cases in TTCN-3
MaTeLo	http://www.all4tec.net	Embedded systems	Enhanced Markov Chains	Probabilities for transitions and inputs	Textual test plans and executable test cases in TTCN-3, etc.
MBTsuite	http://www.smartesting.com	Software	UML State Machines or Activities	Test cases and verification points	Various, i.e., Excel, Selenium, HO Quality Center …

Agenda

- Introduction
- Test automation at a glance
- Models for test automation
- Industrial standards and notations
- **Findings from industry**
  - Is model-based testing already accepted by industry?
  - Challenges and recommendations for MBT integration
  - ROI considerations and improvement potential
  - **Selected tools for industrial application**
- Conclusion and discussion

Conclusion and discussion
To recap

- Automation helps automating clerical tasks in order to gain productivity
  - Needs upstream activities and thorough planning
- Models can be used to increase the degree of software test automation further
- Automated test execution is already established and mature
- Automated test design (manifest as model-based testing) is still not broadly applied
  - Potential is recognized
  - Important industrial standards refer or recommend MBT
- Use of models for testing can also be helpful even if test generation is not employed
- Challenges need to be tackled before MBT can unfold its full power
- Industrial pioneers have shown the applicability, cost saving potential and scalability of MBT approaches

Conclusion and discussion
Quality of test models are essential

- Implicit and imperfect knowledge of the tester and the test basis or made explicit in a test model

- Quality of test models influence the quality of resulting test case specifications

- Test model may vary in terms of
  - Used language and notation
  - Abstractions
  - Abstraction layer

- Test models are usually simpler than the system model/specification of the system under test -> that does not mean that the test model is simple itself

- Appropriate visualization helps to bridge the gap among stakeholders

Thanks for your attention!
Questions?! I’m certain there are some – or even many

References

[Som06] Ian Sommerville: Software Engineering, Sommerville, Ian: Software
Engineering, Addison-Wesley Longman, Amsterdam, 8th edition,
June 4 2006, 8th edition, June 4, 2006.

[Kle03] Kleppe, Anneke: MDA Explained. Addison-Wesley Longman, Amsterdam.
2003. ISBN 978-0321194428

[UMLs11] Object Management Group (OMG): Unified Modeling Language
Superstructure (UMLs), Version 2.4,
http://www.omg.org/spec/UML/2.4.1/, August 2011.

[OMG03] Object Management Group (OMG): MDA-Guide. http://www.omg.org/
cgi- bin/doc?omg/03-06-01.

[Rai04] Raistrick, Chris et al.: Model Driven Architecture with Executable UML.
Cambridge University Press, 2004. ISBN 978-0521537711

[EES11] ETSI ES 202 951: Requirements for Modeling Notations. ETSI Standard,
Methods for Testing and Specification (MTS); Model-Based Testing
(MBT). V1.1.1 (2011-07)

[WiM11] Model-Based Testing: http://en.wikipedia.org/wiki/
Model- based_testing. Last visit: 05.01.2012

[Utt06]: Utting, M.; Pretschner, A., Legeard, B.: A Taxonomy of
Model- Based Testing. ISSN 1170-487X, 2006. URL:
http://www.cs.waikato.ac.nz/pubs/wp/2006/uow-cs-wp-20

References

[ISTQB] International Software Testing Qualifications Board (ISTQB): ISTQB/GTB standard glossary for testing terms. http://www.software-tester.ch/PDF-Files/CT_Glossar_DE_EN_V21.pdf

[Pre] Prenninger W., and Pretschner, A., Abstractions for Model-Based Testing, in *International Workshop on Test and Analysis of Component Based Systems* (TACoS 2004), (Bonn, GER, 2004), Pages 59-7.

[UTP] Object Management Group (OMG): UML Testing Profile. URL: http://www.omg.org/spec/UTP

[Schie] Schieferdecker, I.: Modellbasiertes Testen. OBJEKTSpektrum 3/07, S. 39-45, 2007.

[SwissQ] SwissQ, Trends und Benchmark Report Schweiz, Wo stehen wir, wo geht es hin? Testing 2013, http://de.slideshare.net/swissq/testing-trends-und-benchmarks-2013-web

[Weiss] Weißleder, Stephan: Relation of Model-Based Testing and Safety-Relevant Standards. 1. ETSI MBT User Conference, Berlin, 2011. URL: http://www.model-based-testing.de/mbtuc11/

[Spil] Spillner, A. et al., "Wie wird in der Praxis getestet? Umfrage in Deutschland, Schweiz und Österreich," *ObjektSpektrum,* May 2011 (in German); www.sigs-datacom.de/fileadmin/user_upload/zeitschriften/os/2011/Testing/spillner_vosseberg_OS_testing_11.pdfLast

[Weiss2] Weißleder et al., MODELLBASIERTES TESTEN: HYPE ODER REALITÄT?, ObjektSpektrum 06/2011.

Marc-Florian Wendland

Fraunhofer FOKUS, CC MOTION

Fon: +
49 (0) 30 – 3463 7395

E-Mail:
marc-florian.wendland@fokus.fraunhofer.de

Web:
www.fokus.fraunhofer.de
www.fokusmbt.com