



Modellbasiertes Testen vs. Testgetriebene Modellierung

Dr. Baris Güldali

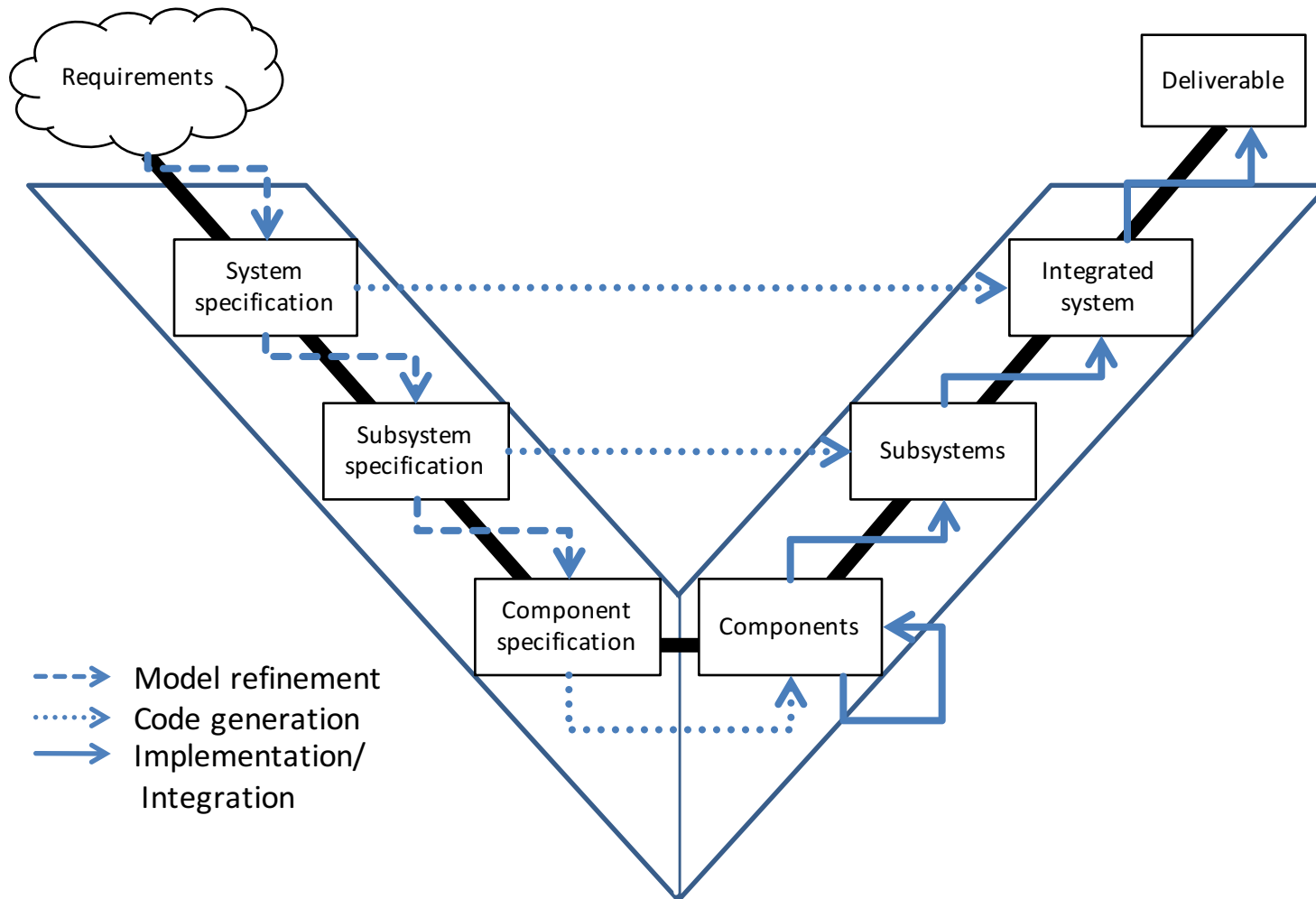
06.10.2015, MESCONF/München



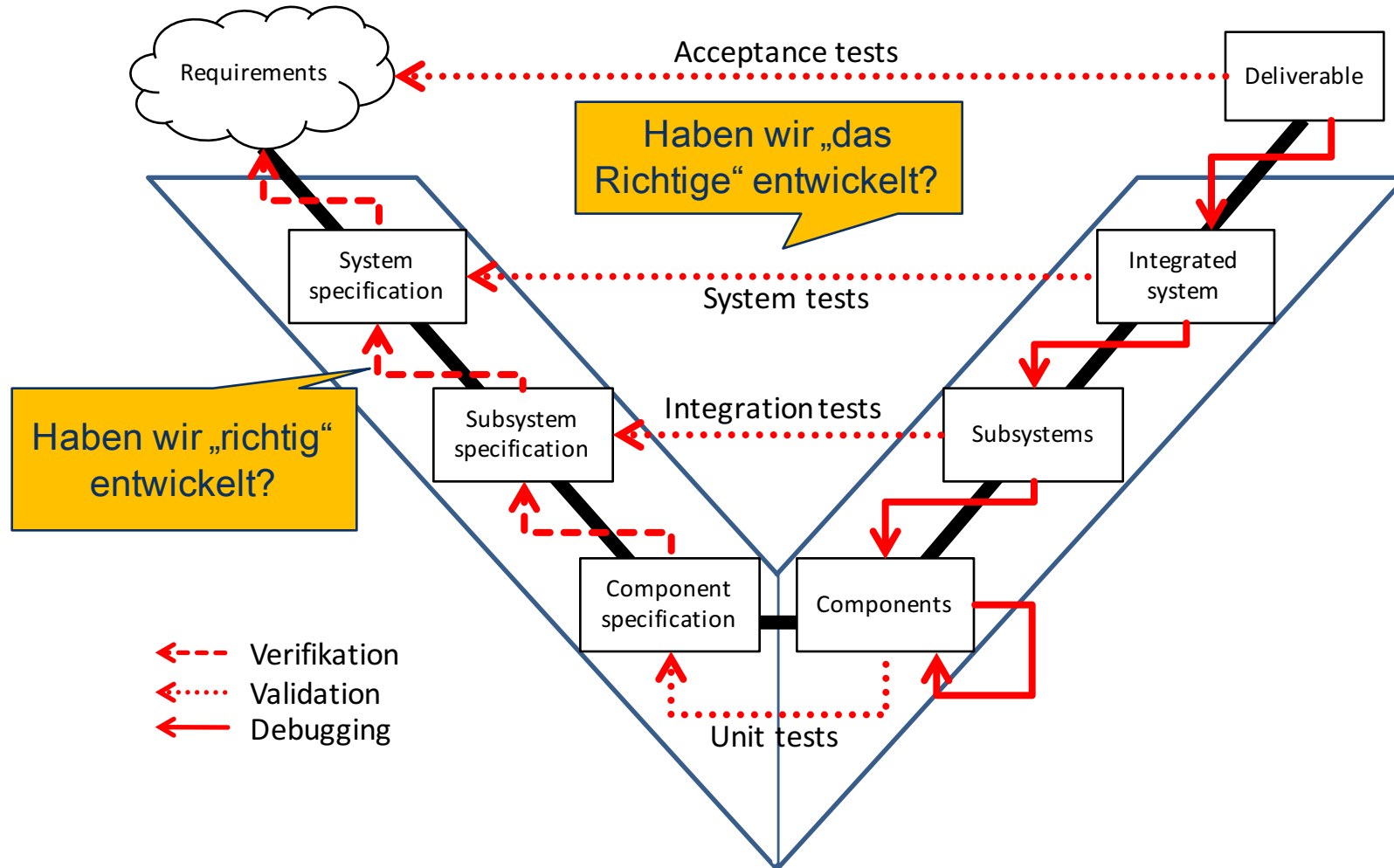
UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft



- Teil 1: Modelle $\leftrightarrow$ Testfälle
- Teil 2: Was bedeutet das Manifest für Tester?

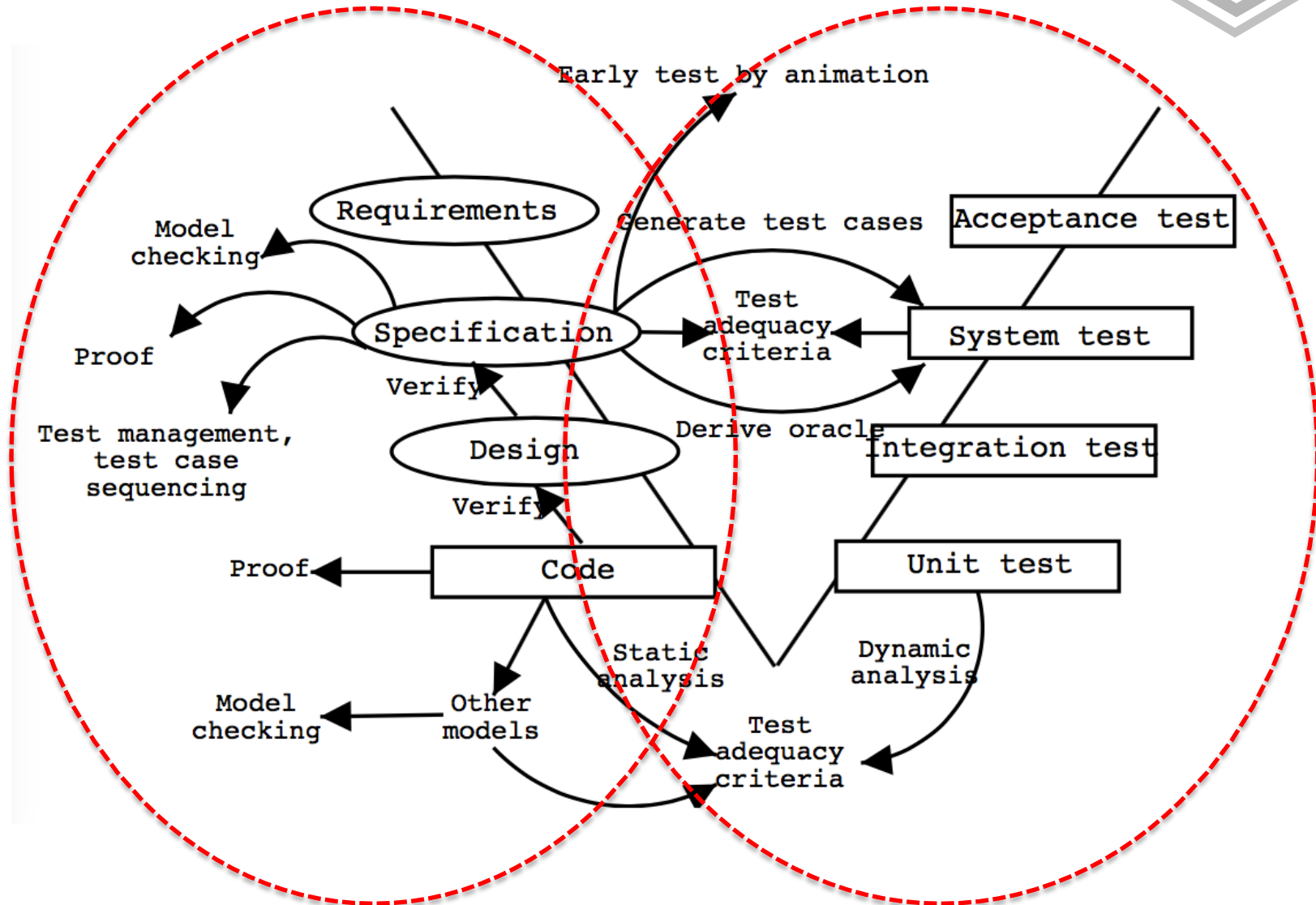


[L. Baresi, M. Pezzè, 2006]

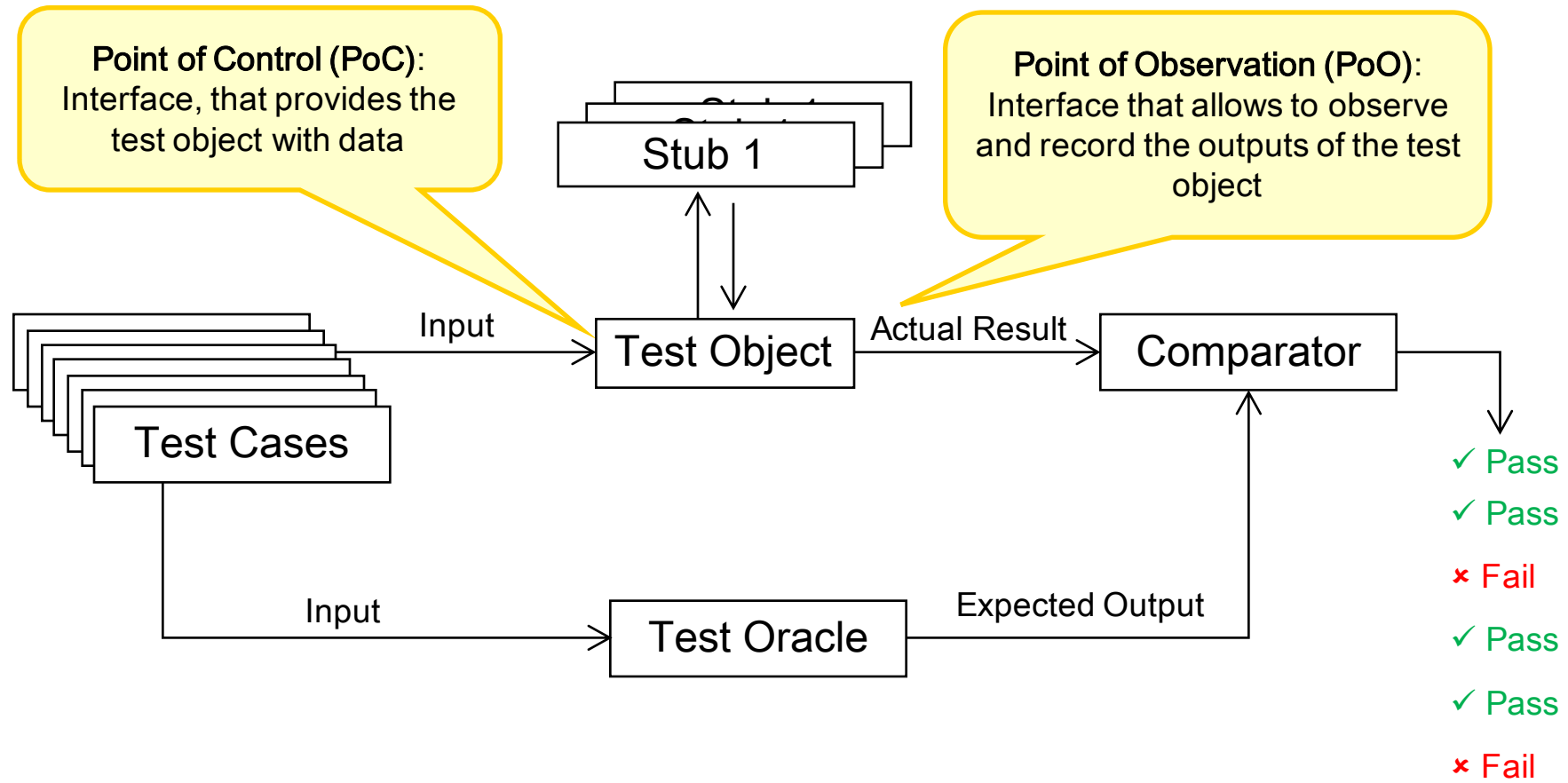


[L. Baresi, M. Pezzè, 2006]

Verifikation vs. Validation



Bagdanov: Working together - Formal Methods and Testing, 2003

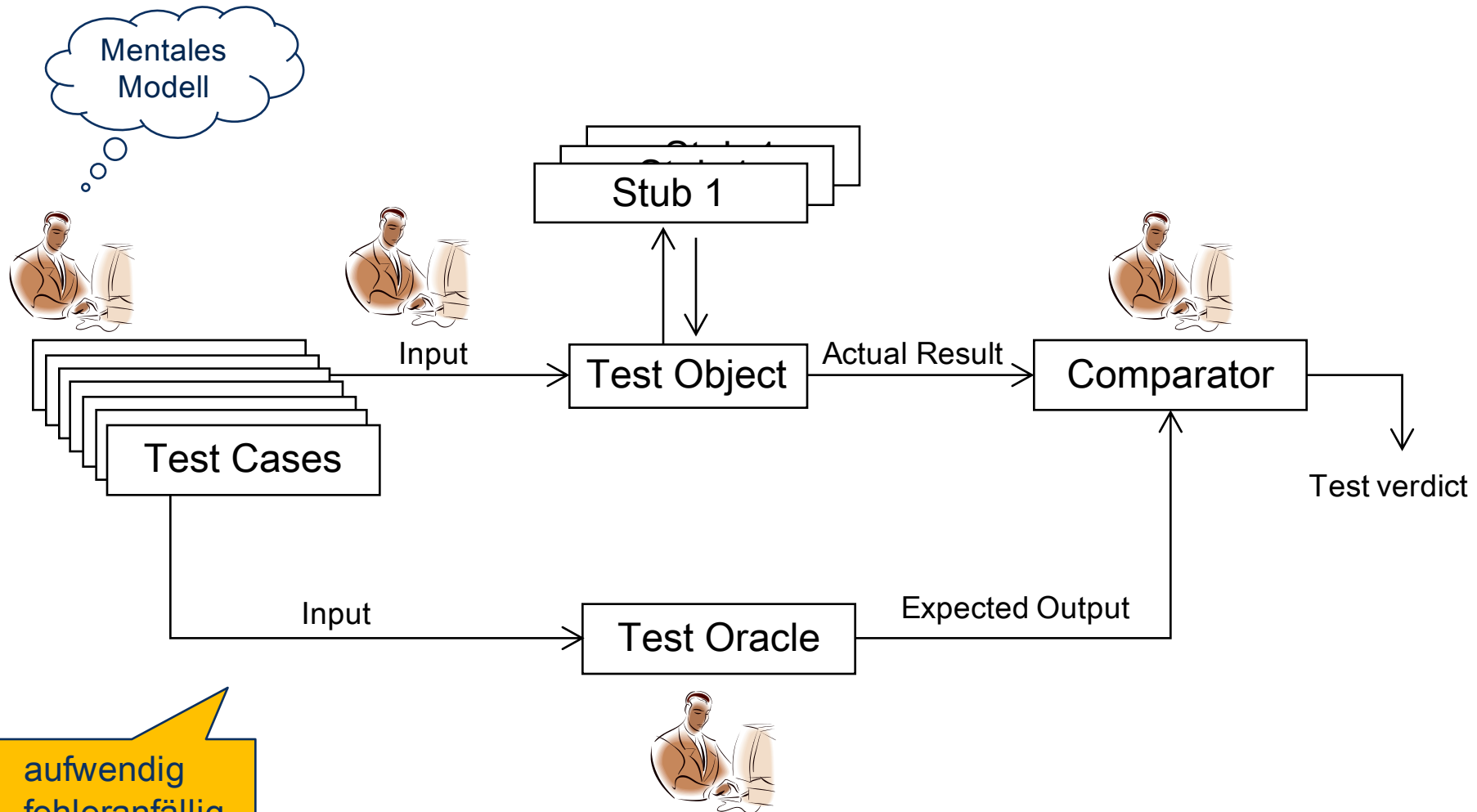


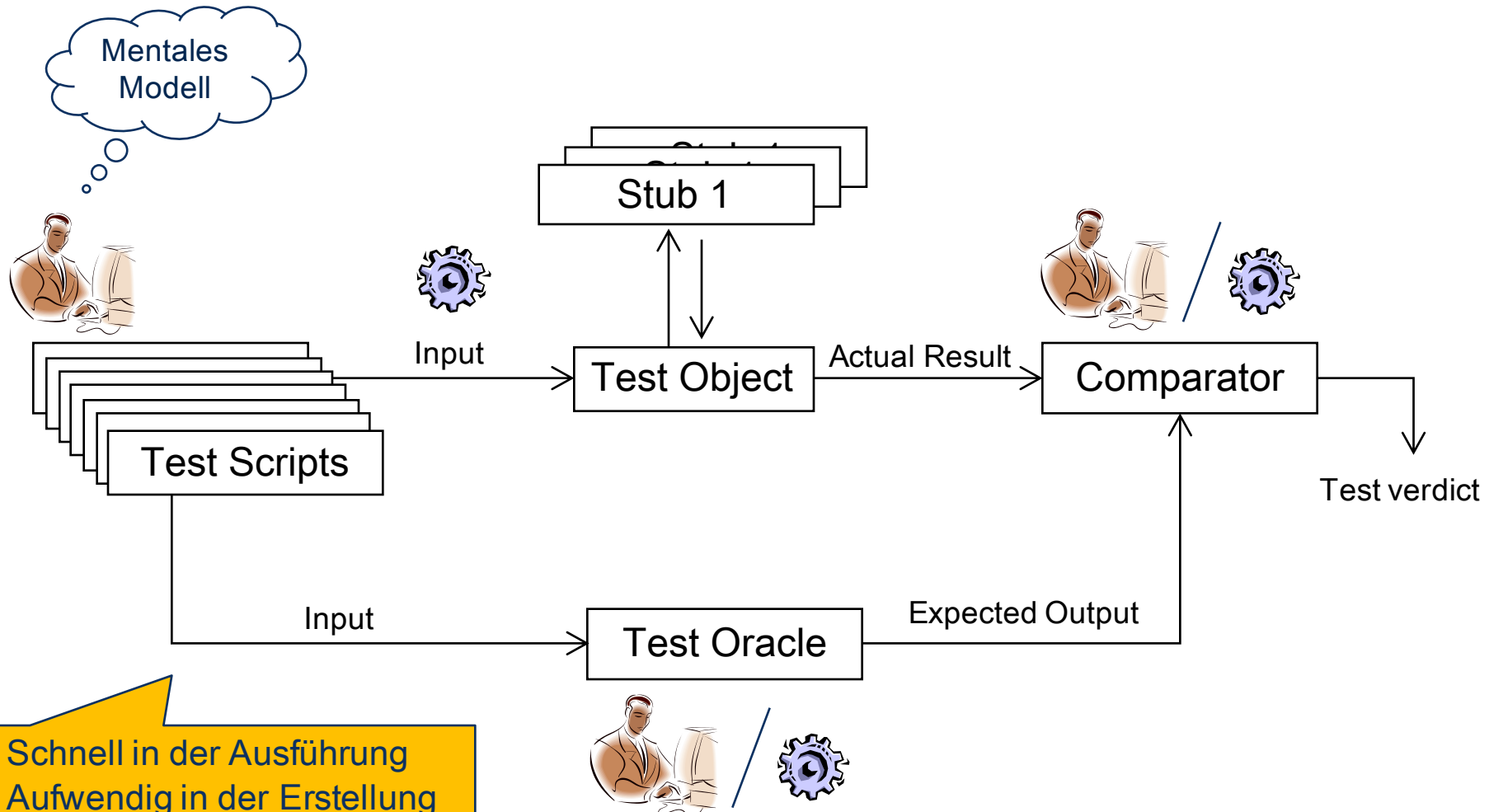
»Program testing can be used to show the presence of bugs, but never to show their absence!«

Edsger W. Dijkstra, 1970

»Testen erhöht das Vertrauen in die Qualität des Systems, wenn wenige oder keine Fehler gefunden werden. «

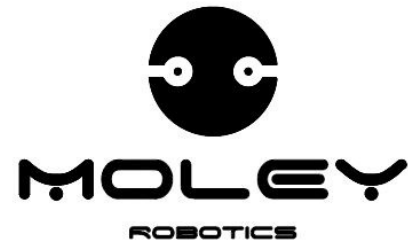
GTB, 2007

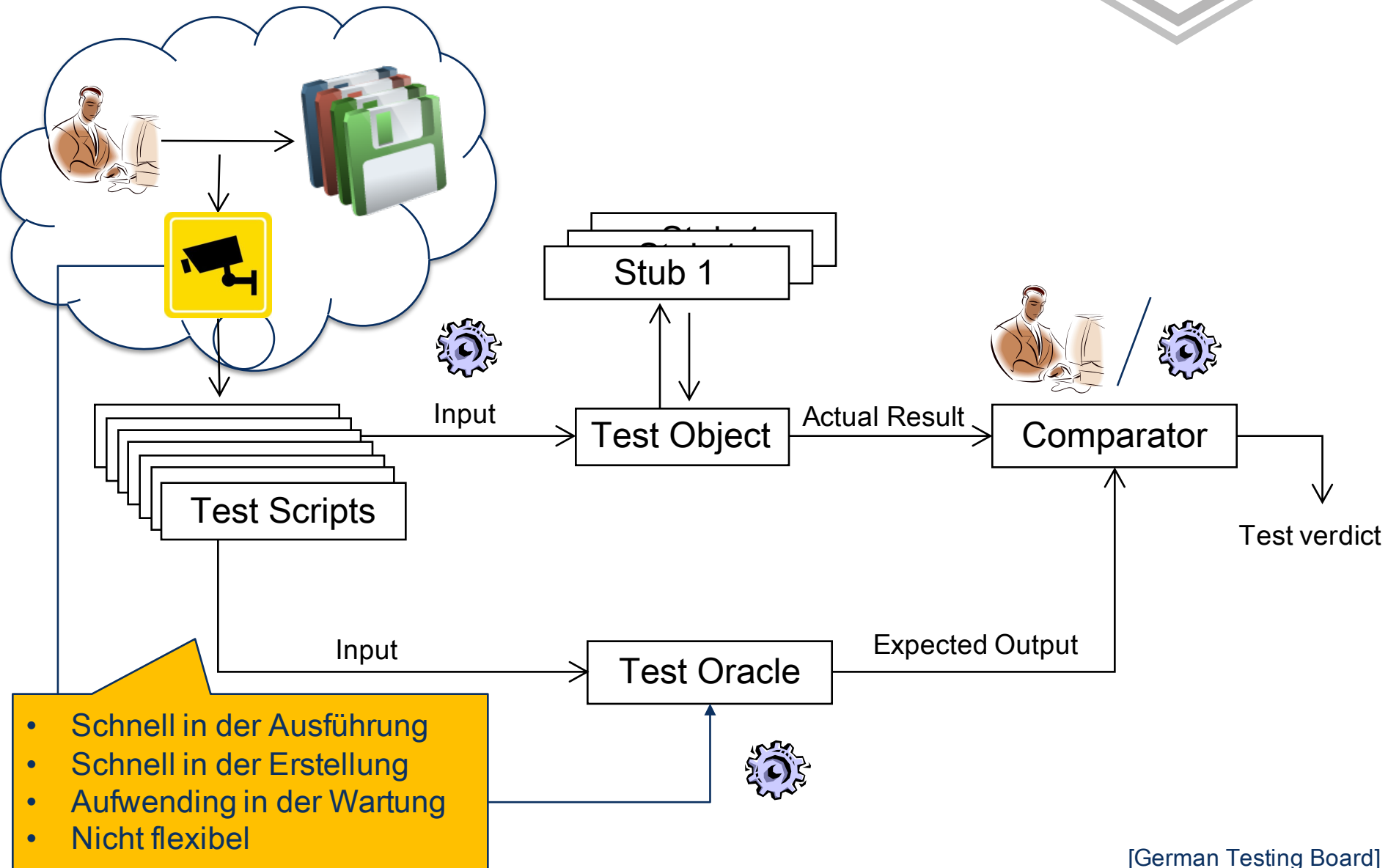




- Schnell in der Ausführung
- Aufwendig in der Erstellung
- Aufwendung in der Wartung

Küchenroboter Moley

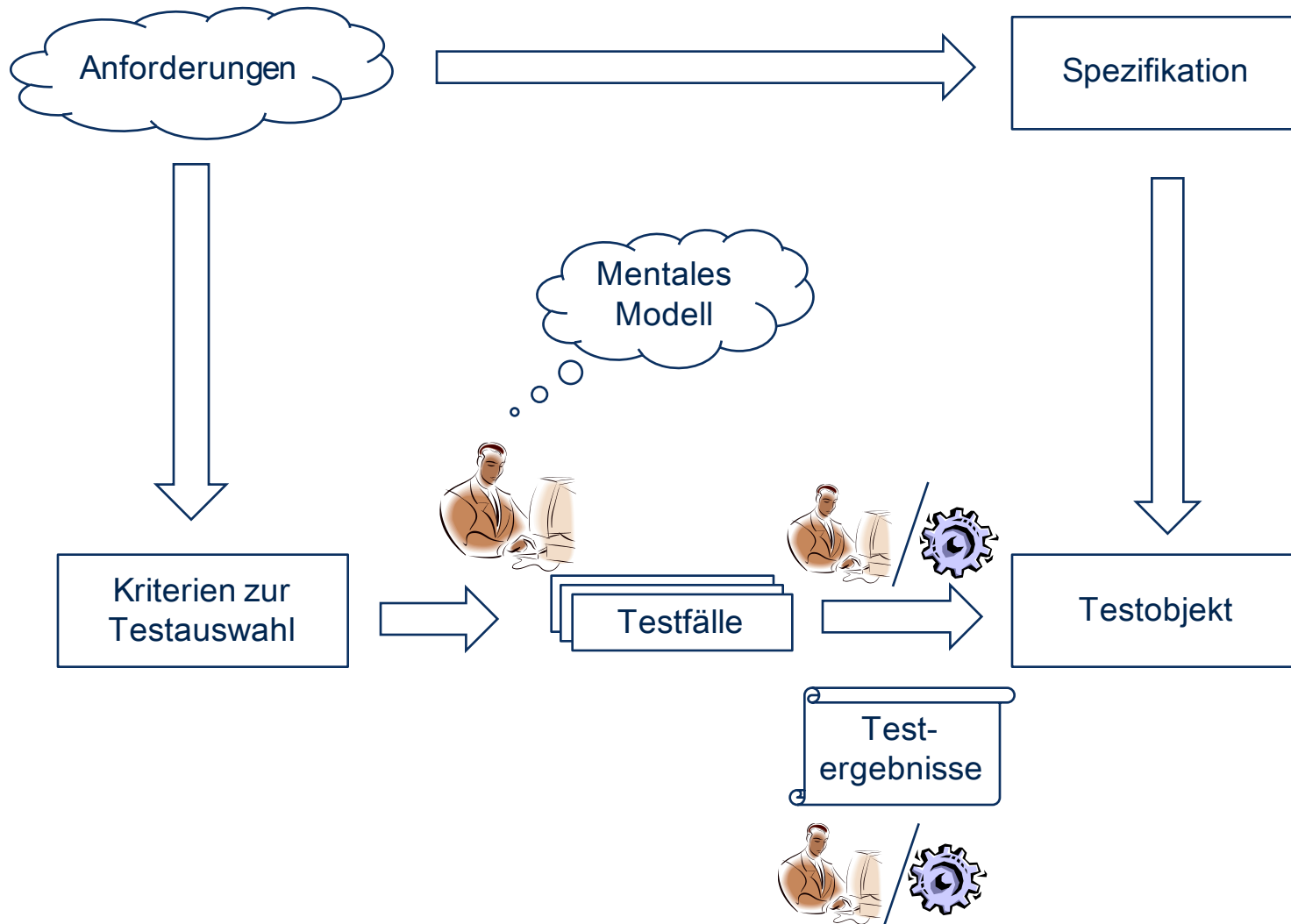


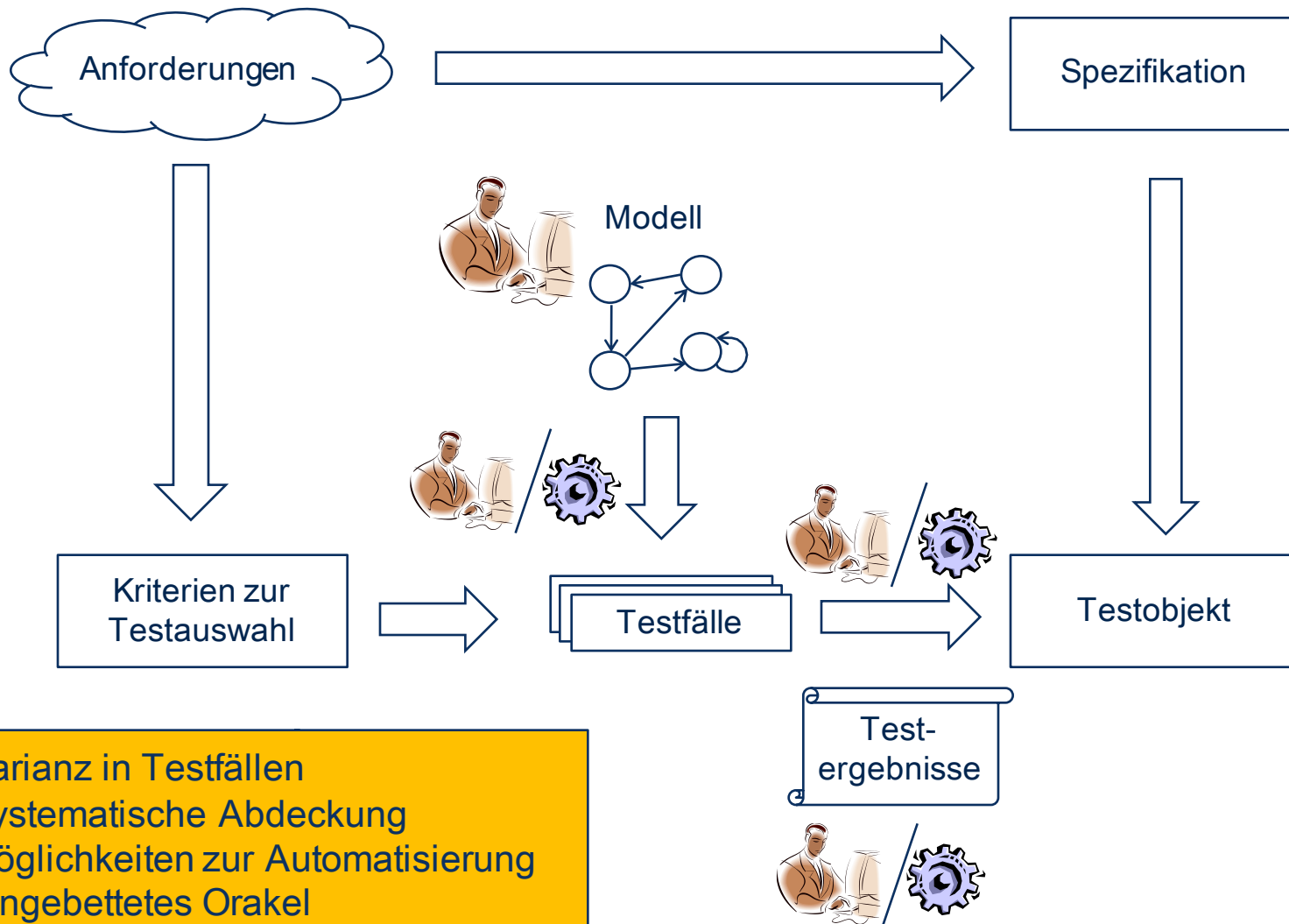


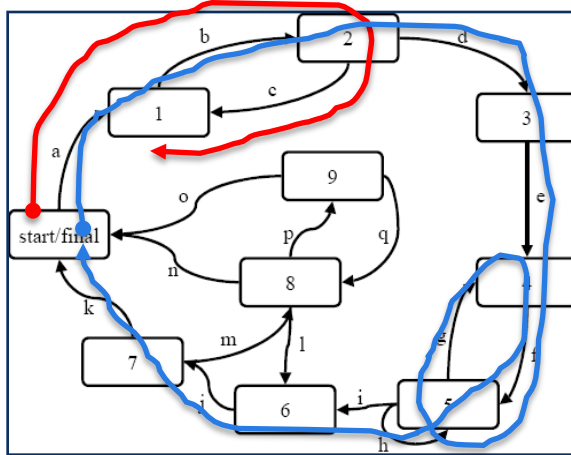
Find the bug!



<https://www.youtube.com/watch?v=rrX9ofGvad0>







Selektionskriterien für Testfälle zur Modellabdeckung

Plattform-unabhängige oder Plattform-spezifische Testfälle

derive

```
Test_Case {
  a → b → c → assert(1)
}
```

```
Procedure Test_Script() {
  //precondition
  set_state(start);

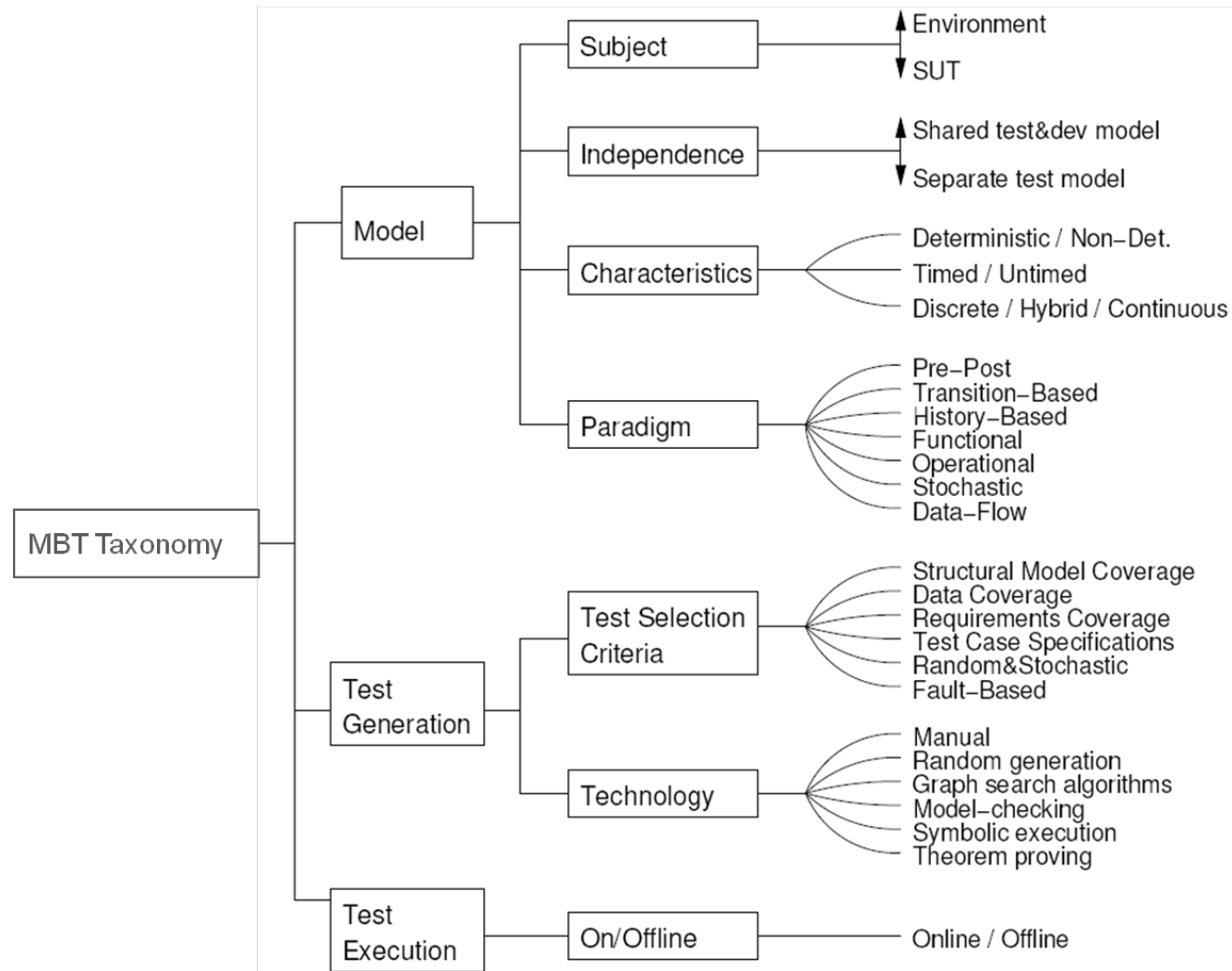
  //body
  a();
  b();
  c();

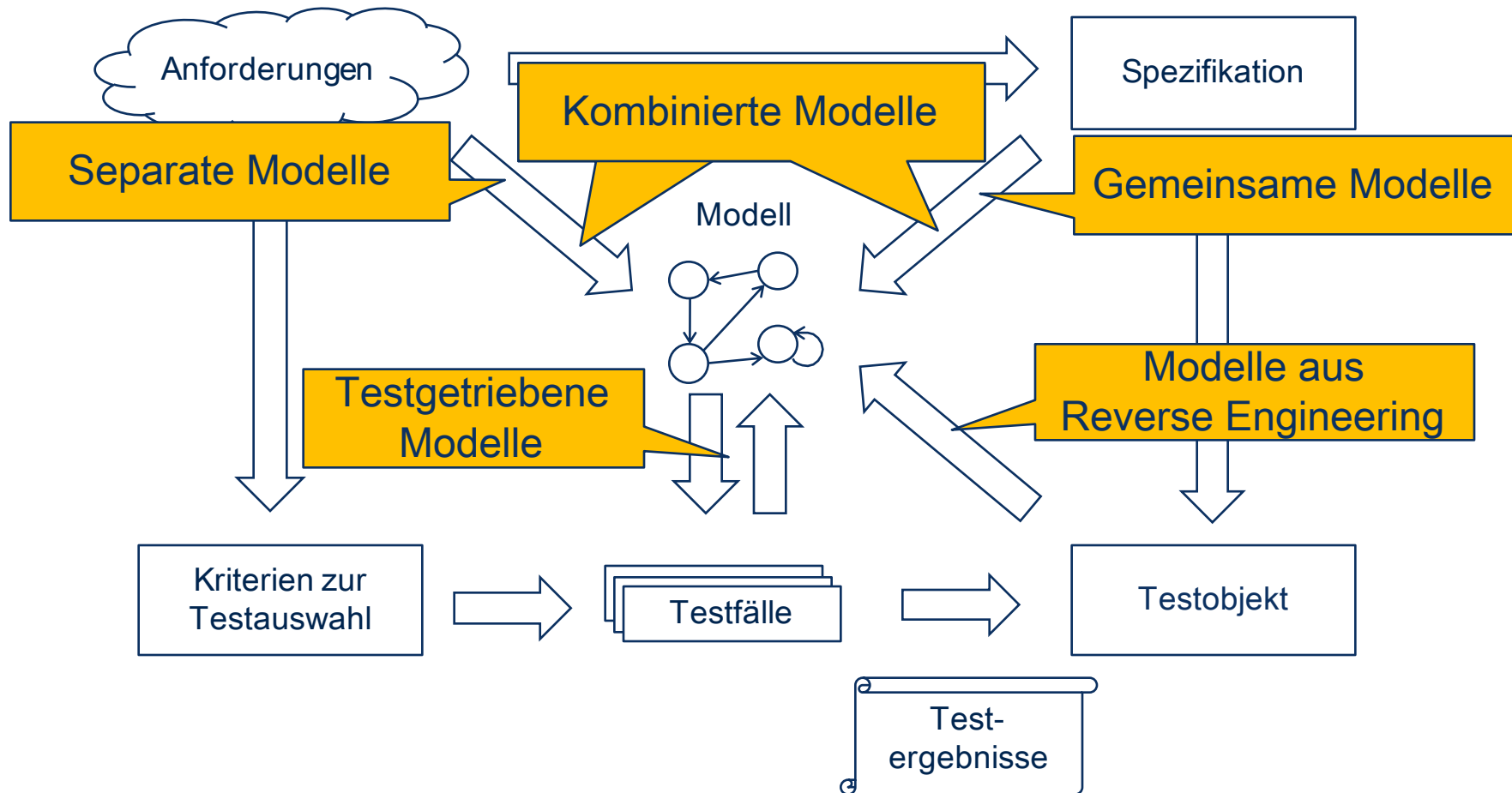
  //post
  assert_state(1);
}
```

invoke

```
var state;

Procedure a() {
  //code for event a
}
Procedure b() {
  //code for event b
}
Procedure c() {
  //code for event c
}
```

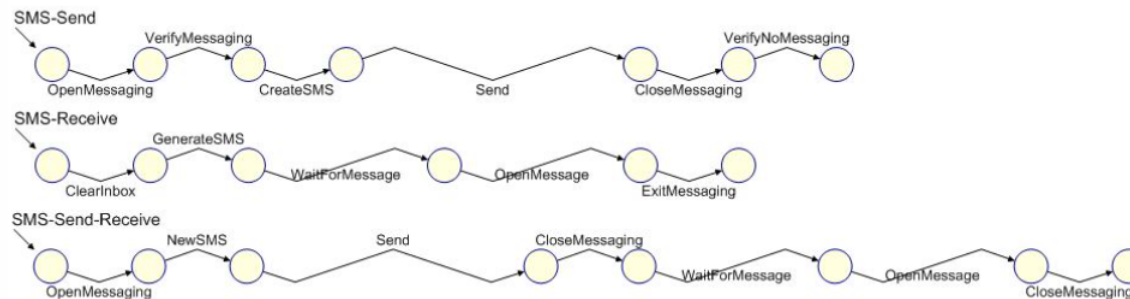




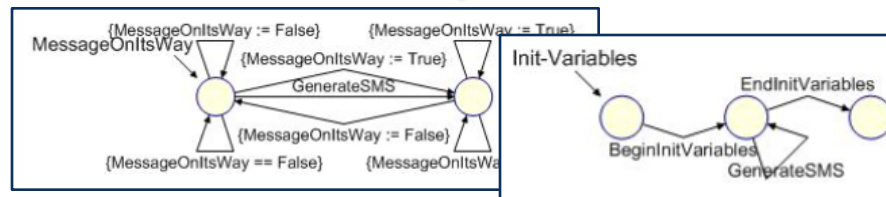
Testgetriebene Modelle



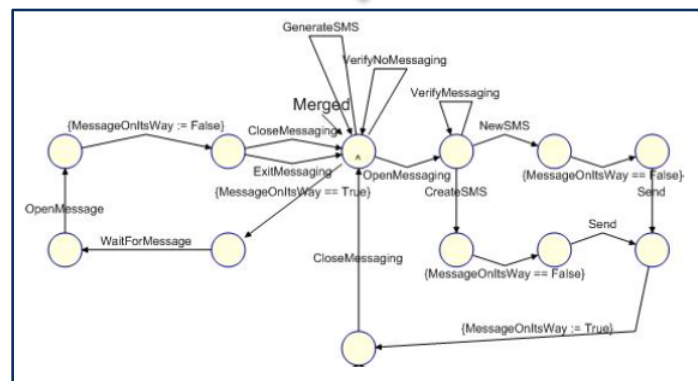
Technische
Universität
Tampere



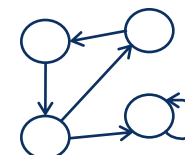
synthesize



merge



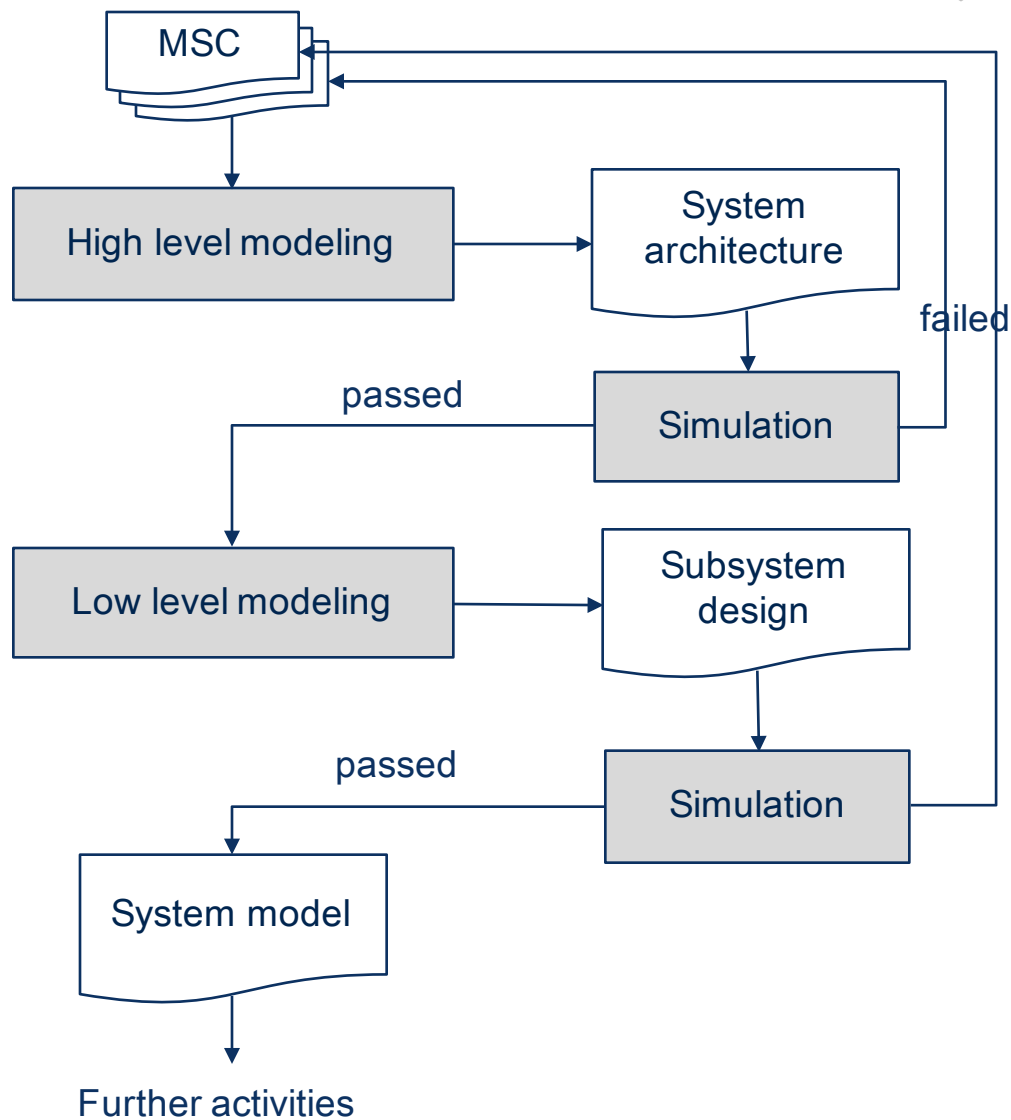
Modell



Testgetriebene Modelle



MOTOROLA



- Modell-orientiertes Testen
 - Modelle als Kommunikationsmittel
 - manueller Testfallentwurf
 - manuelle/automatisierte Testdurchführung möglich
- Modell-getriebenes Testen
 - Transformierbare Modelle
 - automatisierter Testfallentwurf
 - oft automatisierte Testdurchführung
- Modell-zentriertes Testen
 - ausführbare Modelle als zentrale Artefakte
 - durchgehende Automatisierung
 - Testfälle und Testergebnisse sind Sichten des Modells

- Ausprägungen
 - Modell-orientiertes Testen
 - Modell-getriebenes Testen
 - Modell-zentriertes Testen
- Allg. Anforderungen (IEEE 830)
 - Korrektheit
 - Eindeutigkeit
 - Vollständigkeit
 - Konsistenz
 - Gewichtung
 - Validierbarkeit
 - Änderbarkeit
 - Nachverfolgbarkeit
- Testbarkeitsanforderungen (Binder)
 - Steuerbarkeit (PoC)
 - Beobachtbarkeit (PoO)
 - Variabilität
 - Entscheidbarkeit

- Schwergewichtig
 - Hoher initial Aufwand zur Erstellung der Modelle
 - Hoher laufender Aufwand für die Wartung/Anpassung
 - Neue Skills von Tester notwendig
 - Werkzeuge / Werkzeugintegration
- Ein Manifest für MBT fällig? :)

INTEGRIERE STAKEHOLDER

STREBE LANGLEBIGE MODELLEN AN

ERSTELLE VALIDIERBARE, TRANSFORMIERBARE
UND AUSFÜHRBARE MODELLE

FRONT-LOADING

VERMEIDE DUPLIKATION UND UNNÖTIGE
WIEDERHOLUNGEN

MACHE MODELIERUNG LEICHT ZUGÄNGLICH

ETABLIERE EINE MODELLIERUNGSKULTUR

INTEGRIERE STAKEHOLDER

STREBE LANGLEBIGE MODELLEN AN

ERSTELLE VALIDIERBARE, TRANSFORMIERBARE
UND AUSFÜHRBARE MODELLE

FRONT-LOADING

VERMEIDE DUPLIKATION UND UNNÖTIGE
WIEDERHOLUNGEN

MACHE MODELIERUNG LEICHT ZUGÄNGLICH

ETABLIERE EINE MODELLIERUNGSKULTUR

- Tester und Endnutzer
- Sprachen für Tester
 - UML, UTP
 - TTCN-3
 - TDL
 - KDT, BDD

INTEGRIERE STAKEHOLDER

STREBE LANGLEBIGE MODELLEN AN

ERSTELLE VALIDIERBARE, TRANSFORMIERBARE
UND AUSFÜHRBARE MODELLE

FRONT-LOADING

VERMEIDE DUPLIKATION UND UNNÖTIGE
WIEDERHOLUNGEN

MACHE MODELIERUNG LEICHT ZUGÄNGLICH

ETABLIERE EINE MODELLIERUNGSKULTUR

- Testmodelle ersetzen Testsuits
- Ermögliche Regressionstests

INTEGRIERE STAKEHOLDER

STREBE LANGLEBIGE MODELLEN AN

ERSTELLE VALIDIERBARE, TRANSFORMIERBARE
UND AUSFÜHRBARE MODELLE

- Korrektheit
- Vollständigkeit
- Konsistenz
der Modelle

FRONT-LOADING

VERMEIDE DUPLIKATION UND UNNÖTIGE
WIEDERHOLUNGEN

MACHE MODELIERUNG LEICHT ZUGÄNGLICH

ETABLIERE EINE MODELLIERUNGSKULTUR

INTEGRIERE STAKEHOLDER

STREBE LANGLEBIGE MODELLEN AN

ERSTELLE VALIDIERBARE, TRANSFORMIERBARE
UND AUSFÜHRBARE MODELLE

FRONT-LOADING

Modelle und Testentwurf
unmittelbar nach
Anforderungsdefinition

VERMEIDE DUPLIKATION UND UNNÖTIGE
WIEDERHOLUNGEN

MACHE MODELIERUNG LEICHT ZUGÄNGLICH

ETABLIERE EINE MODELLIERUNGSKULTUR

INTEGRIERE STAKEHOLDER

STREBE LANGLEBIGE MODELLEN AN

ERSTELLE VALIDIERBARE, TRANSFORMIERBARE
UND AUSFÜHRBARE MODELLE

FRONT-LOADING

VERMEIDE DUPLIKATION UND UNNÖTIGE
WIEDERHOLUNGEN

- Modell-getriebenes Testen
- Modell-zentriertes Testen

MACHE MODELIERUNG LEICHT ZUGÄNGLICH

ETABLIERE EINE MODELLIERUNGSKULTUR

INTEGRIERE STAKEHOLDER

STREBE LANGLEBIGE MODELLEN AN

ERSTELLE VALIDIERBARE, TRANSFORMIERBARE
UND AUSFÜHRBARE MODELLE

FRONT-LOADING

VERMEIDE DUPLIKATION UND UNNÖTIGE
WIEDERHOLUNGEN

MACHE MODELIERUNG LEICHT ZUGÄNGLICH

ETABLIERE EINE MODELLIERUNGSKULTUR

- Übernehme Werkzeuge der Entwickler
- Integration mit Testwerkzeugen

INTEGRIERE STAKEHOLDER

STREBE LANGLEBIGE MODELLEN AN

ERSTELLE VALIDIERBARE, TRANSFORMIERBARE
UND AUSFÜHRBARE MODELLE

FRONT-LOADING

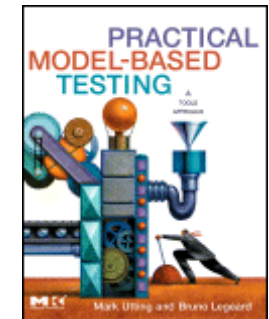
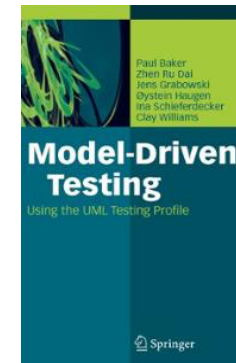
VERMEIDE DUPLIKATION UND UNNÖTIGE
WIEDERHOLUNGEN

MACHE MODELIERUNG LEICHT ZUGÄNGLICH

ETABLIERE EINE MODELLIERUNGSKULTUR

- “Was du nicht modellieren kannst, kannst du nicht Testen”
- ISTQB-Schulung für MBT

- Einstieg
 - Objektspektrum
 - Buch: Basiswissen MBT
- Expert
 - Buch: Practical MBT
 - Buch: Model-Driven Testing



“It is far better to improve the **effectiveness** of testing first than to improve the **efficiency** of poor testing. Automating chaos just gives faster chaos.”

(Fewster 1999)

Softwaretest in Praxis und Forschung



Umfrage 2015

Zu den rollenbezogenen
Fragebögen der Umfrage 2015:

Vom 15.9.-15.12.2015

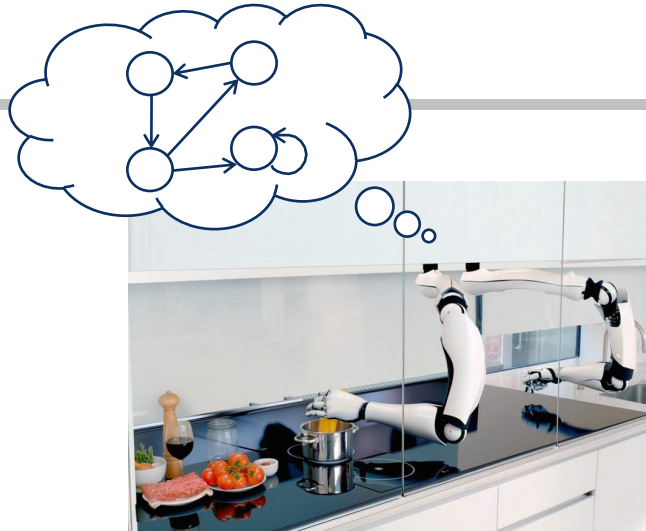
Management >

Operativ >

Forschung >



Teilnahme bis 15. Dezember 2015



Vielen Dank für Ihre Aufmerksamkeit!

Dr. Baris Güldali
s-lab – Software Quality Lab
Universität Paderborn
Zukunftsmeile 1, 33102 Paderborn
Tel.: (05251) 605392
bguldali@s-lab.upb.de
<http://s-lab.upb.de>

MBT Werkzeuge: Beispiele



Testfallgenerator	Unterstützter Modelltyp	Hersteller
TestBench	UML	imbus
TestDesigner	UML	SmartTesting
Qtronic	UML + Java/C#	Conformiq
Reactis	Simulink/Stateflow	Reactive System
Rational Statemate/ Rhapsody	UML, SysML	IBM
MBTsuite	UML	AFRA/sepp.med

Liste evtl. veraltet