Manuelles Testen mit TSM

Stuttgarter Test-Tage am 16. & 17. April 2015

Dr. Rainer Schmidberger
Tobias Hirning

rainer.schmidberger@informatik.uni-stuttgart.de

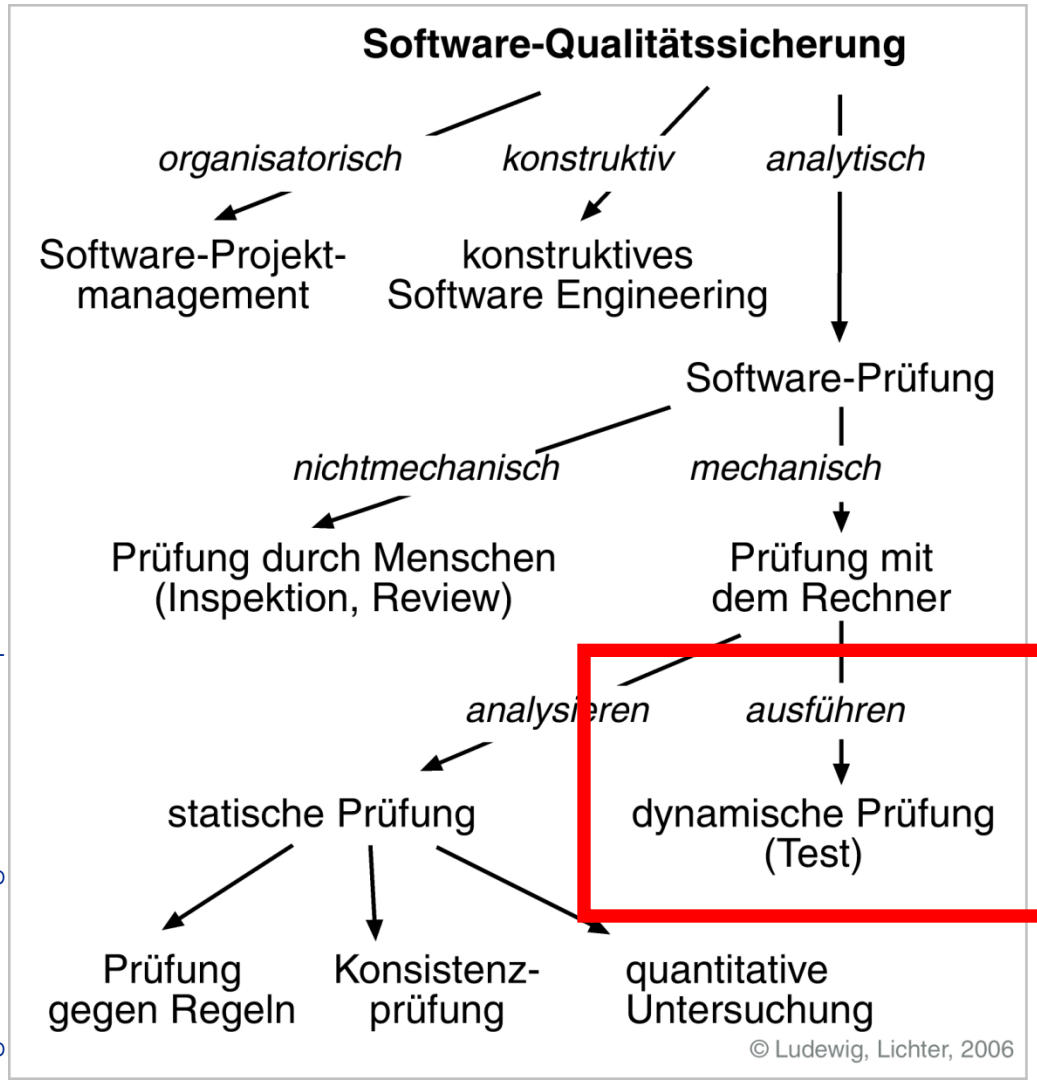
Manuelles Testen mit TSM

Agenda

- Kurze Einführung, Begriffe
- Grundprinzipien von TSM
- TSM am praktischen Beispiel
- Diskussion

Quellen:
Einige der Folien basieren auf den Materialien von Prof. Dr. Jochen Ludewig und Prof. Dr. Martin Glinz

Prüfverfahren



„Testen ist die Ausführung eines Programms mit der Absicht, Fehler zu finden“

[Myers, 1979]

und

Dokumentation der genauen Ausführungs-bedingungen wie z.B. Version, Testfälle und Resultate.

Ludewig, Lichter 2010: Software Engineering, dpunkt Verlag

Test

*An activity in which a system or component is **executed** under specified conditions, **the results are observed** or recorded, and an **evaluation is made** of some aspect of the system or component.*

IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology

Automatisch ausgeführter Test

- Effizienz bei häufiger Ausführung
- „Austesten“ – Durchlaufen von Wertebereichen ist möglich
- Reproduzierbarkeit
- Viele gute Tools sind verfügbar, auch Open-Source Tools

- Probleme mit Falsch-Positiven
- Programmierung und Wartung aufwändig

vs.

Manuell ausgeführter Test

- Schnell einsetzbar, kann direkt vom Fachbereich durchgeführt werden, keine „Starthürde“
- Hohe Effektivität: Erkennen von Fehlwirkungen, die nicht zum spezifizierten Soll-Resultat gehören
- „Unempfindlich“ gegenüber Oberflächenänderungen
- Fehlerwirkungen sind in der Regel unmittelbar fachlich plausibel

- Aufwändig und Tester-Kompetenz muss vorhanden sein

Ad-hoc Test

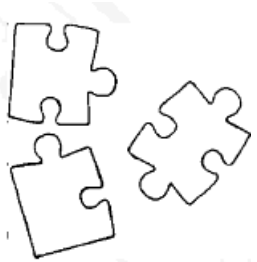
- Die Testziele sind unklar.
- Es ist viel zu oft unklar, welches Systemverhalten als korrekt oder als falsch gilt.
- Die Testergebnisse sind nicht reproduzierbar.
- Der Aufwand für den Test kann vorab nicht geschätzt werden.
- Der Nutzen des Tests ist stark von der Expertise des Tester abhängig.
- Im Schadensfall drohen Haftungsrisiken.

vs. Systematischer Test

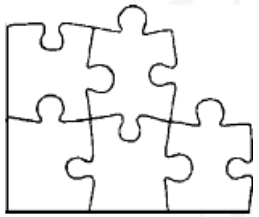
- Test ist geplant, eine Testvorschrift liegt vor, die Testziele (z. B. die Vollständigkeit) sind definiert
- Das Programm wird gemäß Testvorschrift – der Test-spezifikation – ausgeführt. Damit ist der Test jederzeit reproduzierbar und nach-vollziehbar.
- Testergebnisse werden systematisch dokumentiert.
- Der Aufwand für ein Testziel kann prognostiziert werden.
- Der systematische Test lässt sich auch systematisch verbessern.

Teststufen - Übersicht

- **Komponententest**, Modultest, Unit-Test
  → „autarke“ Komponenten werden isoliert getestet



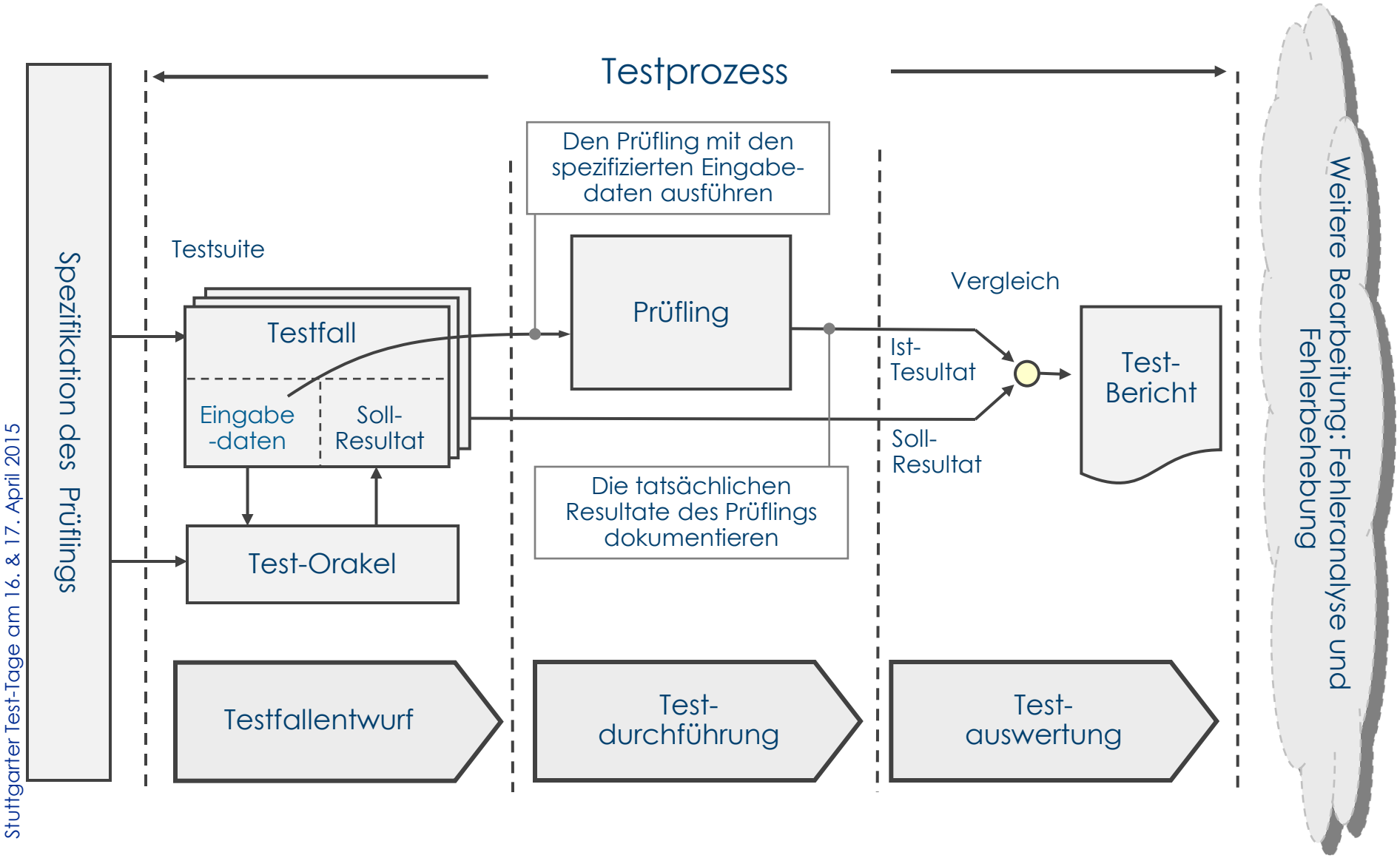
- **Integrationstest**
  → Teilintegrierte Komponenten testen die Komponenteninteraktion



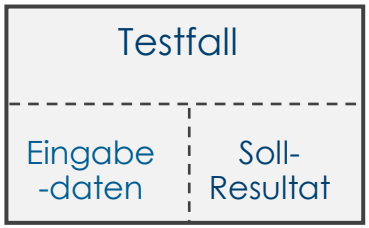
- **Systemtest**
  → Test des Gesamtsystems gegen die Systemspezifikation

Bilder: Martin Glinz, Universität Zürich

Fundamentaler Testprozess



Was ist ein Testfall?

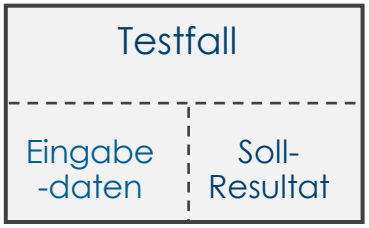


***Testfall** (nach [ISTQB]) umfasst folgende Angaben:*

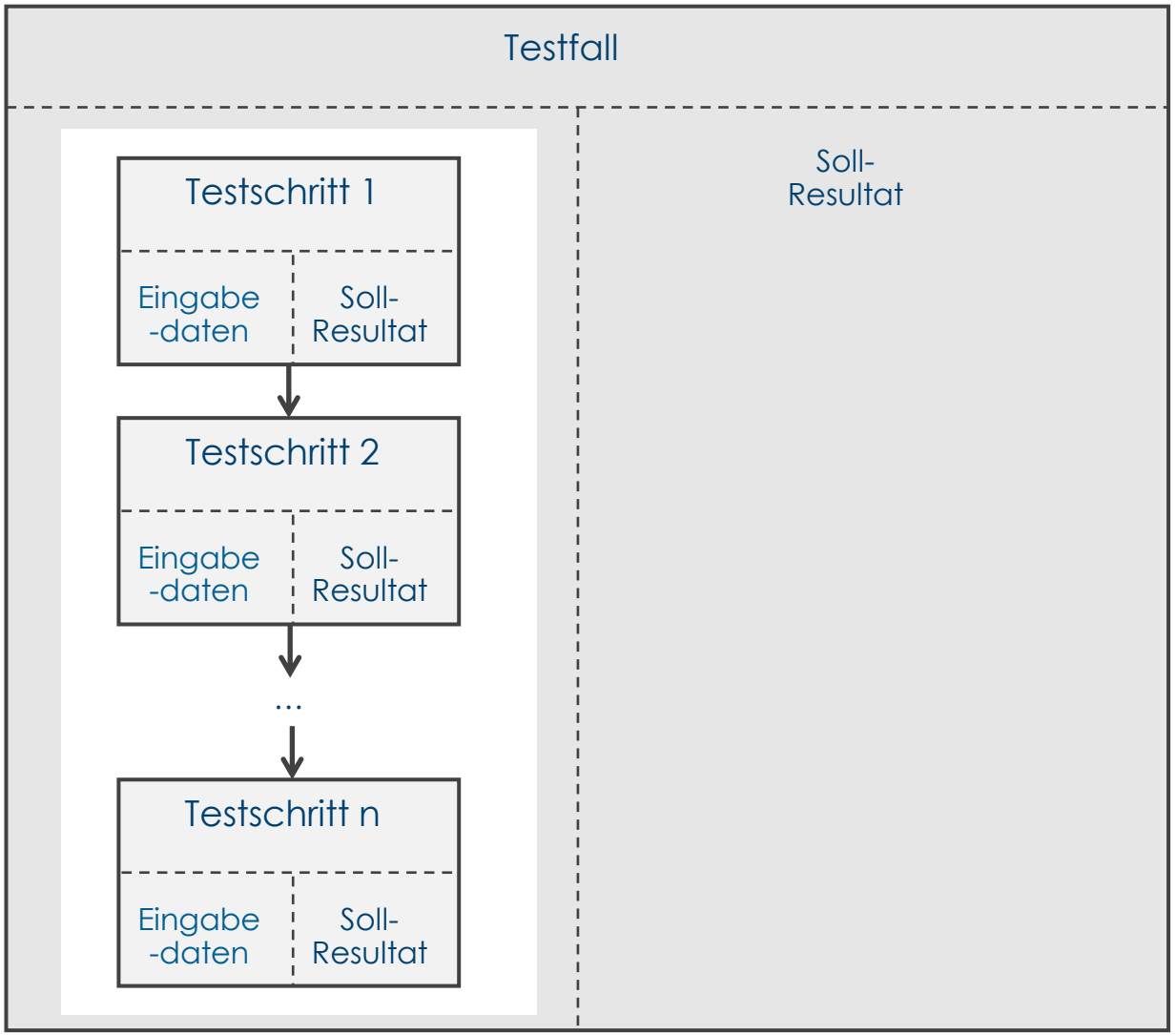
- *die für die Ausführung notwendigen **Vorbedingungen**,*
- *die Menge der **Eingabewerte** (ein Eingabewert je Parameter des Testobjekts),*
- *die Menge der vorausgesagten **Ergebnisse**, sowie die erwarteten Nachbedingungen.*

Testfälle werden entwickelt im Hinblick auf ein bestimmtes Ziel bzw. auf eine Testbedingung, wie z.B. einen bestimmten Programmpfad auszuführen oder die Übereinstimmung mit spezifischen Anforderungen zu prüfen (wie Eingaben an das Testobjekt zu übergeben und Sollwerte abzulesen sind) [nach IEEE 610].

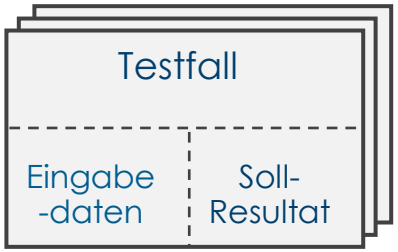
„monolithischer“ Testfall



Testfall bestehend aus Testschritten



Testsuite



Testsuite *(nach [ISTQB]): Die* ***Zusammenstellung*** *(Aggregation)* ***mehrerer Testfälle*** *für den Test einer Komponente oder eines Systems, bei der Nachbedingungen des einen Tests als Vorbedingungen des folgenden Tests genutzt werden können.*

Der „ideale“ Testfall

- Ein Testfall ist gut, wenn er mit hoher Wahrscheinlichkeit einen noch nicht entdeckten Fehler aufzeigt.
- Ein idealer Testfall ist
  - ⇨ **Repräsentativ**: Er steht stellvertretend für viele andere Testfälle
  - ⇨ **Fehlersensitiv**: Er hat nach der Fehlertheorie eine hohe Wahrscheinlichkeit, einen Fehler aufzuzeigen
  - ⇨ **Redundanzarm**: Er prüft nicht, was auch andere Testfälle schon prüfen

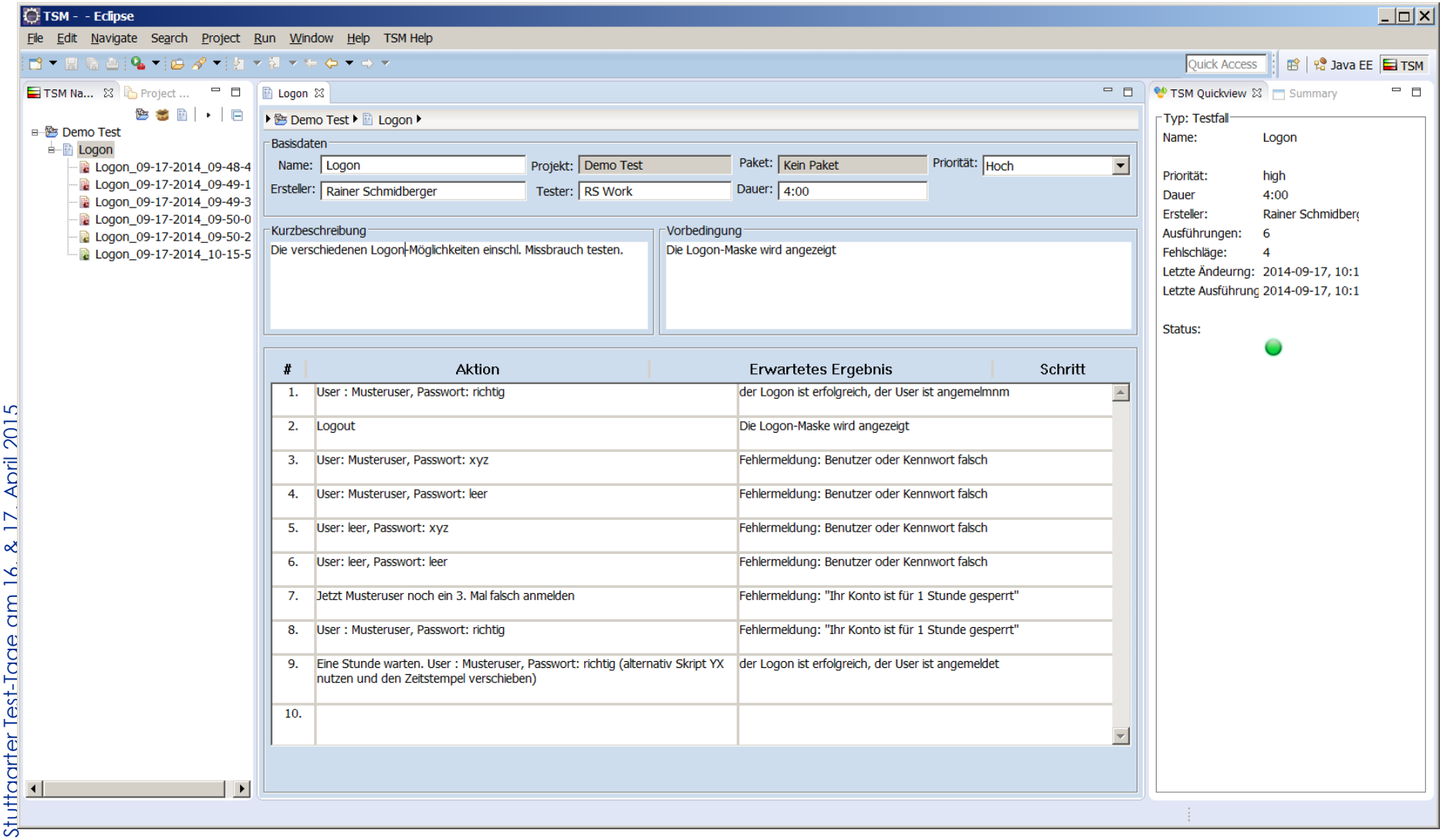
Testfallspezifikation nach [IEEE829]

- **Eindeutige Kennung** jedes Testfall
- **Testgegenstände**: Funktionen bzw. Eigenschaften, die von einem Testfall überprüft werden.
- **Spezifikation der Eingaben**: Detaillierte Angabe aller Eingaben und deren Beziehungen (zum Beispiel (die zeitliche) Abfolge).
- **Spezifikation der Ausgaben**: Auflistung aller erwarteten Ausgaben und weiterer Eigenschaften (wie z.B. Antwortzeiten), die zu erfassen sind und deren Werte bzw. Wertebereiche (Toleranzen)
- **Umgebungsanforderungen**: Detaillierte Beschreibung der Hardware und Softwareanforderungen und anderer Anforderungen. Definition aller speziellen Beschränkungen oder Eigenheiten
- **Wechselbeziehungen** zu anderen Testfällen: Auflistung aller Testfälle, die vor diesem Testfall auszuführen sind und Angabe der Art der Wechselbeziehungen

Testspezifikation, weitere Attribute

- **Priorität**, z. B. beim selektiven Regressionstest
- **Aufwand**: geschätzte Ausführungsdauer, Aufwand für den Test-Setup
- **Autor**: Zuständigkeiten, Tester
- **Fehlersensitivität**: wie oft hat dieser Testfall schon Fehler angezeigt?
- **Ausführungen**: Wann und von wem wurde der Testfall ausgeführt? Und mit welchen Resultaten?
- **Einordnung** nach Funktionsgruppen, Subsystem o. ä.

Das Werkzeug TSM



Das Werkzeug TSM

- TSM ist ein Testwerkzeug, das den manuellen Test unterstützt:
  - ⇨ Erfassen der Testfallspezifikation
  - ⇨ Ausführen der Testfälle und protokollieren der Ergebnisse
  - ⇨ Reporting
- TSM wurde 2012 im Rahmen eines Studienprojekts an der Universität Stuttgart entwickelt und steht unter EPL Lizenz.
- TSM ist derzeit als Eclipse-Plugin oder als „Stand-alone“ RCP-Anwendung verfügbar.
- Ein web-Frontend wird aber Frühjahr 2015 verfügbar sein
- http://tsmtest.sourceforge.net

Live Demo

Tobias Hirning

Manuelles Testen mit TSM

Dr. Rainer Schmidberger
Tobias Hirning

rainer.schmidberger@informatik.uni-stuttgart.de