

Automatisierter Testzyklus für Chipkarten

Thomas Freitag

achelos GmbH

SmartCard-Workshop 2012





Übersicht

1. Einführung / Motivation
2. Historie des Testens
3. Schnittstellen im Testbereich
4. Eclipse Plugins
5. Automatisierung, Harmonisierung
6. Beispiele zur Automatisierung
7. Ausblick

Warum muss Chipkarten Software überhaupt getestet werden?

- Korrekte Funktionalität muss sichergestellt werden
- Konformität zu Spezifikationen und Standards
- Interoperabilität zu Schnittstellen (Kartenleser)
- Schwierige Updatemöglichkeit nach Auslieferung
- Einsatz meistens in sicherheitsrelevanten Bereichen

Phase 1: Manuelles Testen der Gutfälle mit Echtdaten

- manuelles Testen, ohne technische Unterstützung
- nur Gutfälle mit Echtdaten
- Entwickler = Tester



Problem: Echtdaten zu testen reicht nicht aus!



Phase 2: Konstruierte positive und negative Testfälle

- teilweise scriptbasierte automatisierte Testfälle
- konstruierte Testfälle für positive und negative Bereiche
- Entwickler = Tester

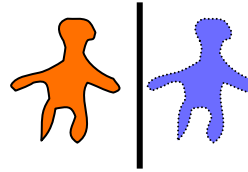


Problem: Entwickler und Tester in einer Person



Phase 3: **Scriptbasiertes Testen auf Basis der Systemspezifikation**

- **scriptbasierte automatisierte Testfälle**
- **Systemspezifikation ist die Basis**
- **Entwicklungsaktivitäten und Testaktivitäten werden getrennt**
- **Entwickler \neq Tester**

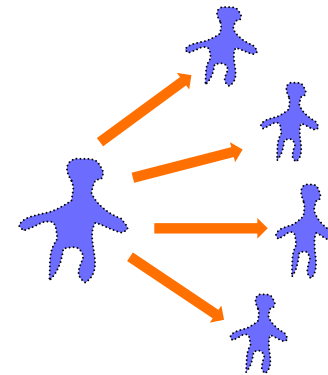


Problem: Keine Aussage über Abdeckung aller UseCases und Anforderungen



Phase 4: Anforderungsgetriebenes Testen

- Testanalyse startet bei den UseCases
- UseCases bzw. Anforderungen sind die Basis der Testfälle
- objektorientierte, automatisierte Testfälle
- Einführung weiterer Rollen im Testbereich





Vorteile einer Kopplung von Anforderungen und Testfällen:

Überprüfbarkeit

- Sind alle Anforderungen abgedeckt?
- Insbesondere wichtig bei Zertifizierungen.

Rückverfolgbarkeit

- Welche Anforderung ist durch einen Fehler verletzt?

Unterstützung von Änderungen

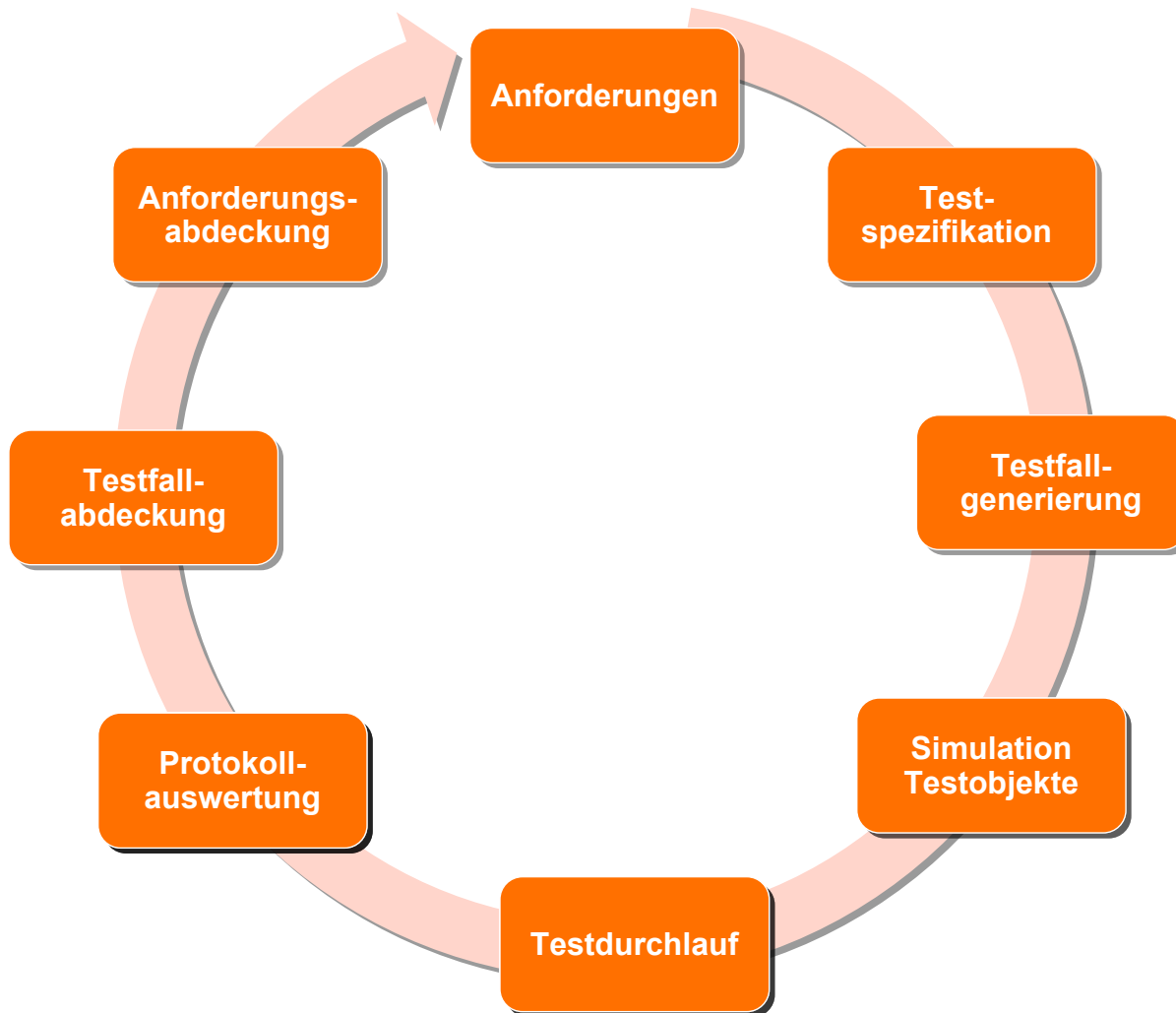
- Welche Testfälle sind bei Änderungen erneut durchzuführen oder anzupassen?

Effizientere Testfallentwicklung

- Genaue Abdeckung der Anforderungen durch Testfälle

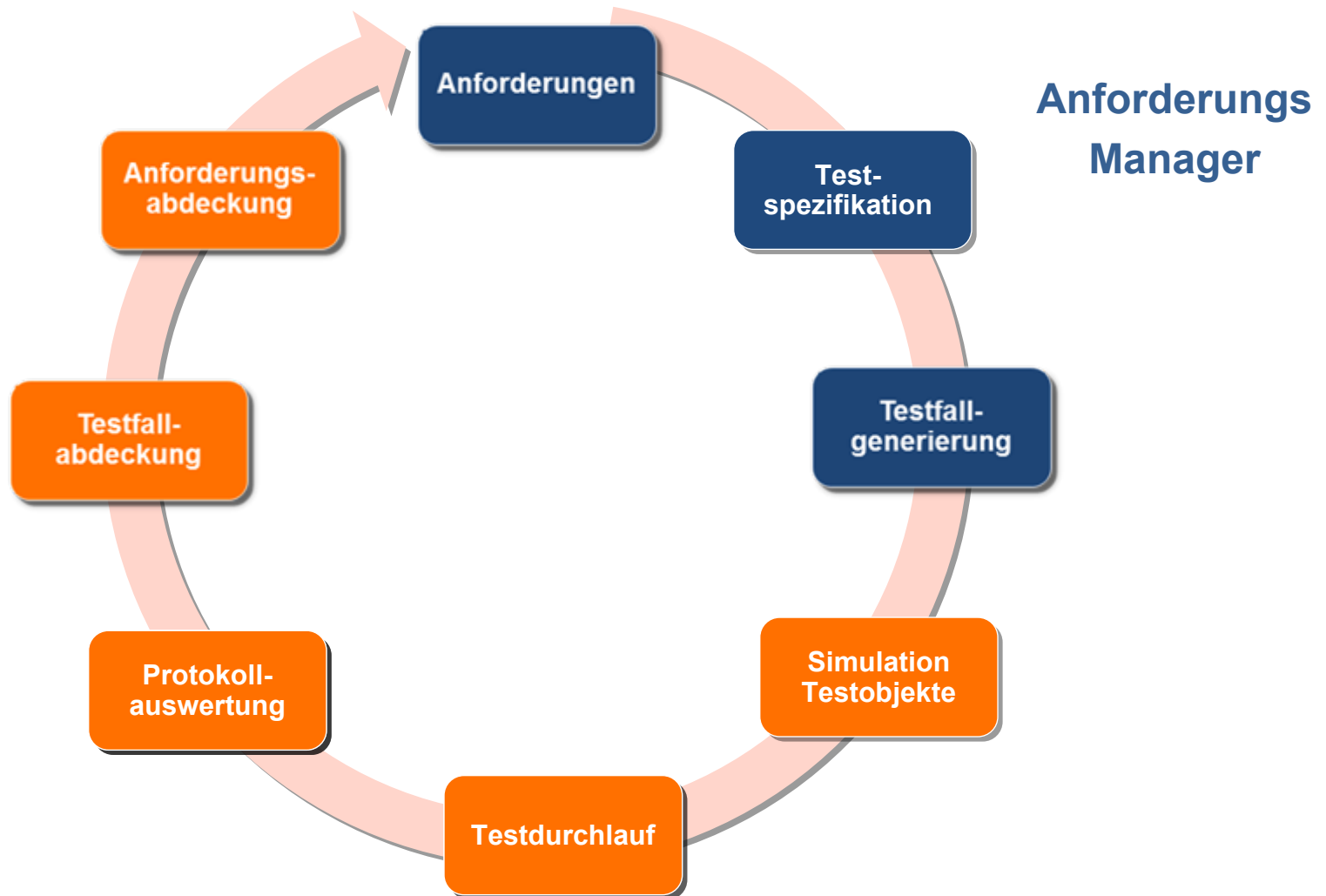


Anforderungsgetriebener Testzyklus



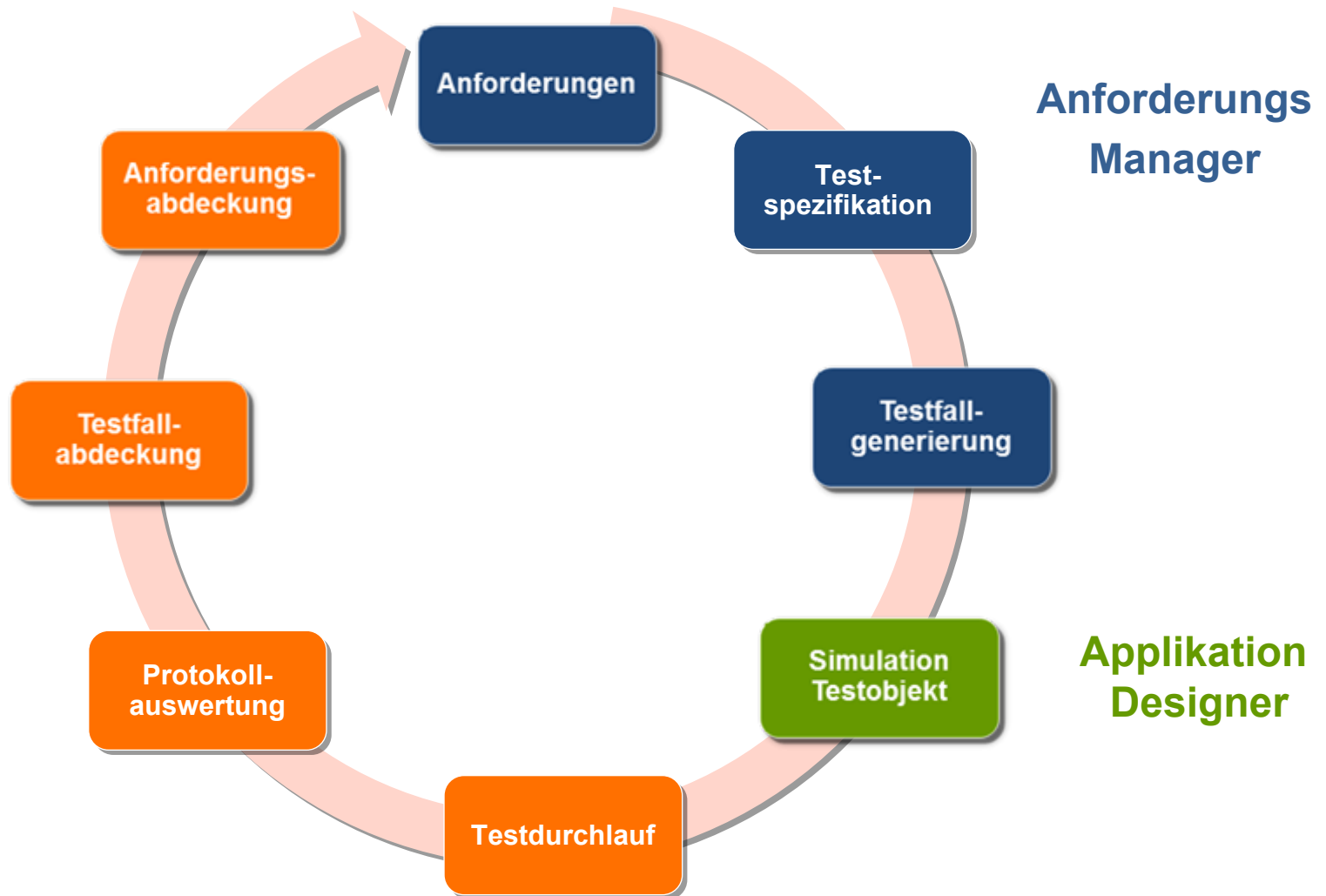


Anforderungsgetriebener Testzyklus (Rollenaufteilung)



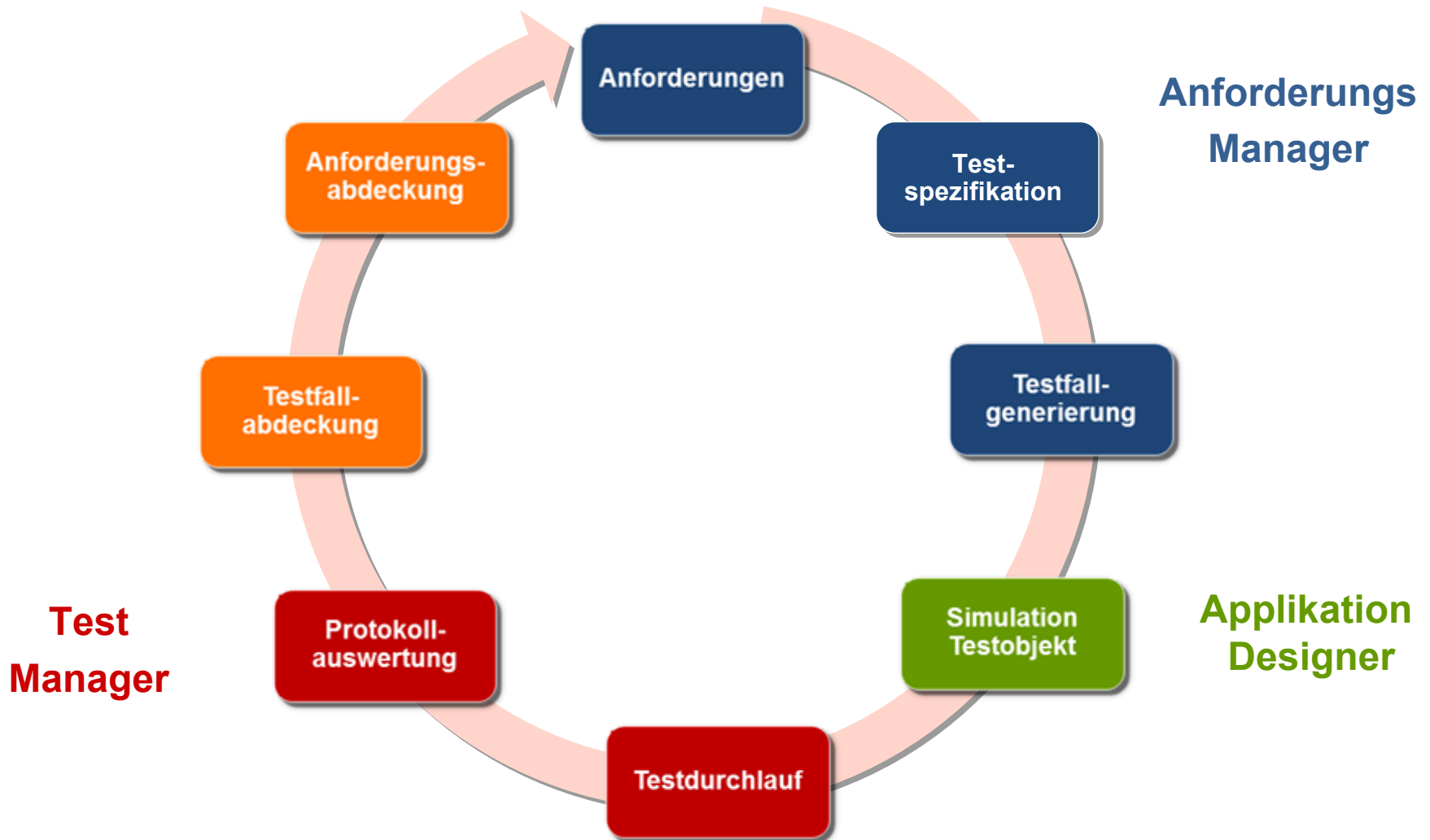


Anforderungsgetriebener Testzyklus (Rollenaufteilung)



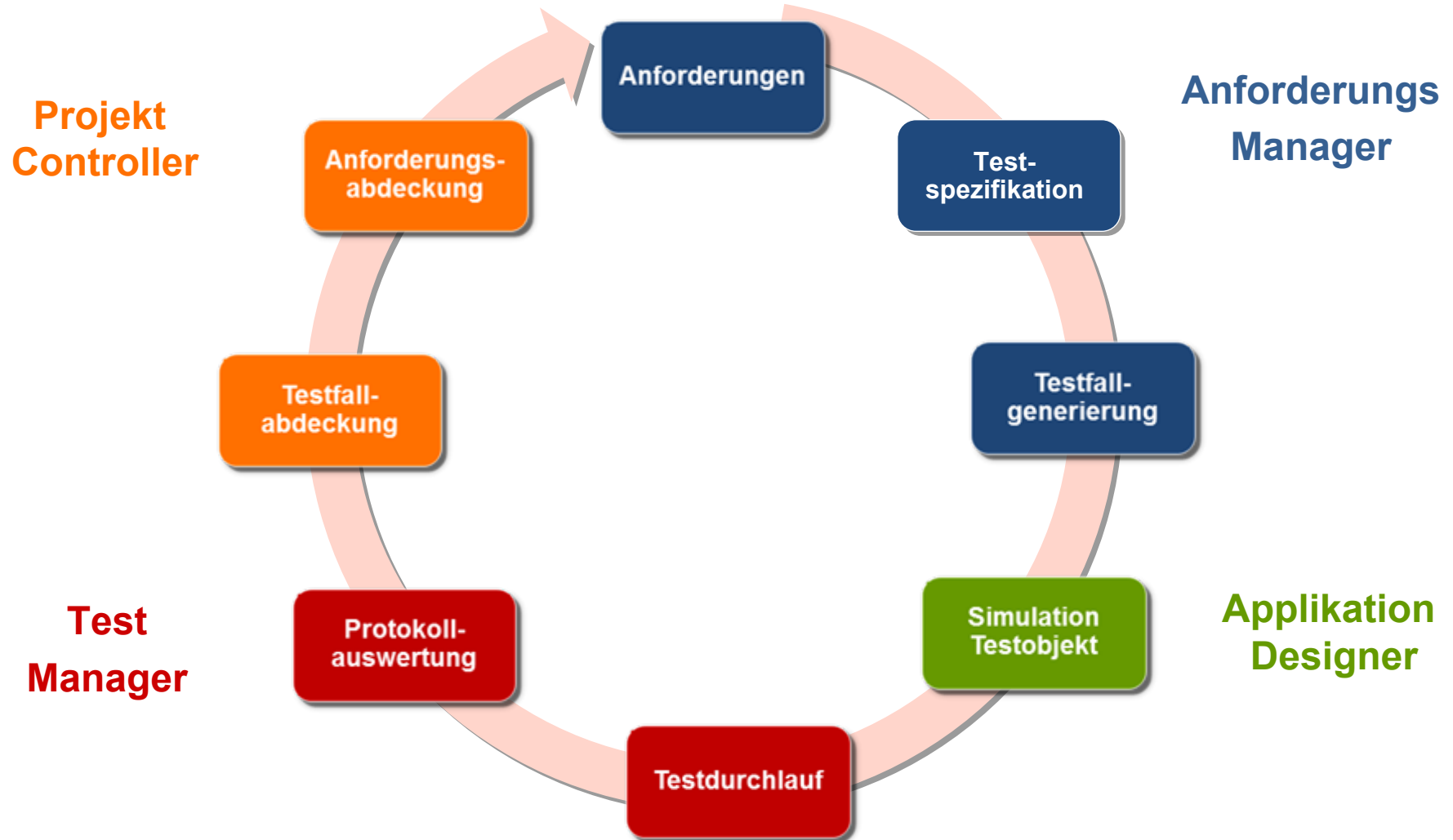


Anforderungsgetriebener Testzyklus (Rollenaufteilung)





Anforderungsgetriebener Testzyklus (Rollenaufteilung)



Schnittstellen im Testbereich

Softwarekonfigurationsmanagement / Versionskontrollsystem

- Ein- und Auschecken von benötigten Dateien

Bug Tracking Tool

- Festhalten und zuweisen von Fehlern

Kartenleser

- 1 zu n Beziehung sinnvoll

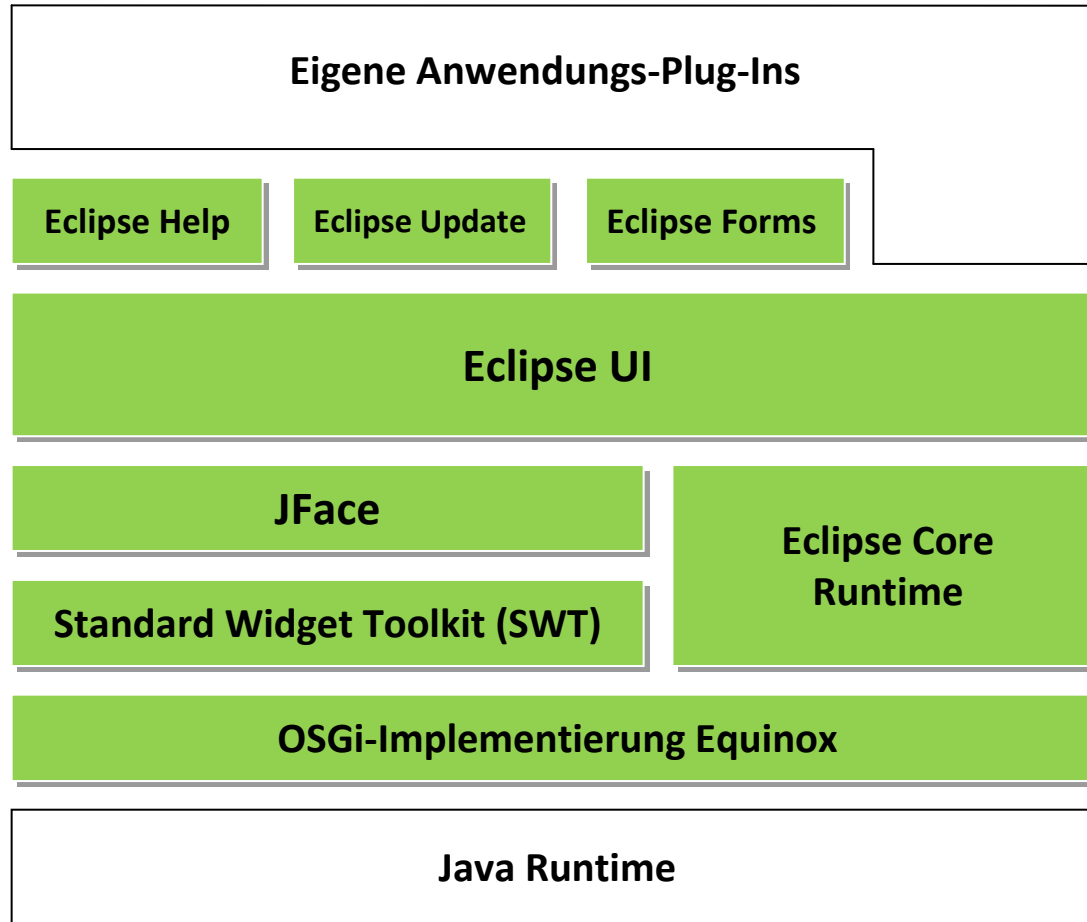
Simulation

- Steuerung, Speicheranipulation

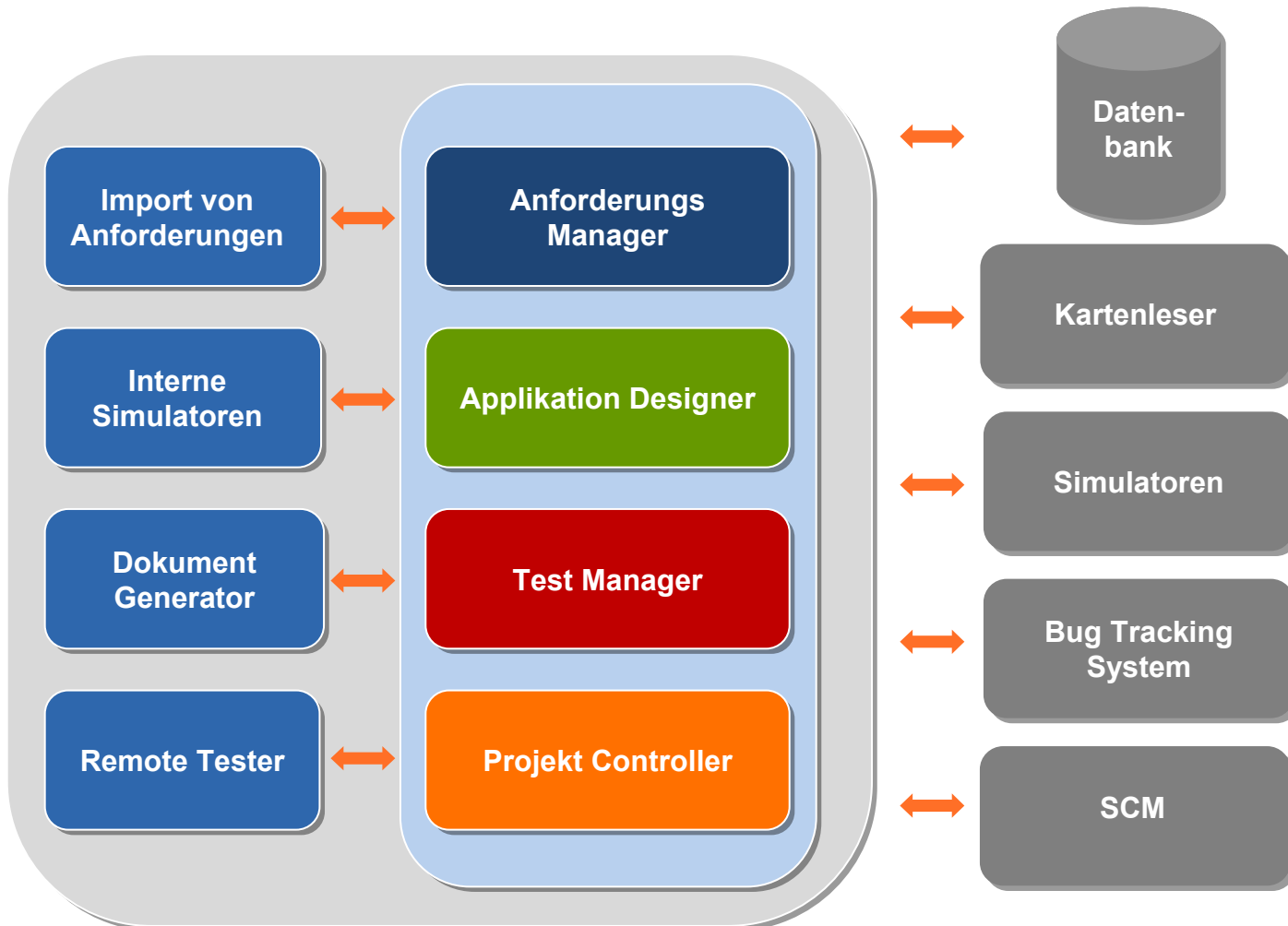
Datenbank

- Ablageort aller Daten

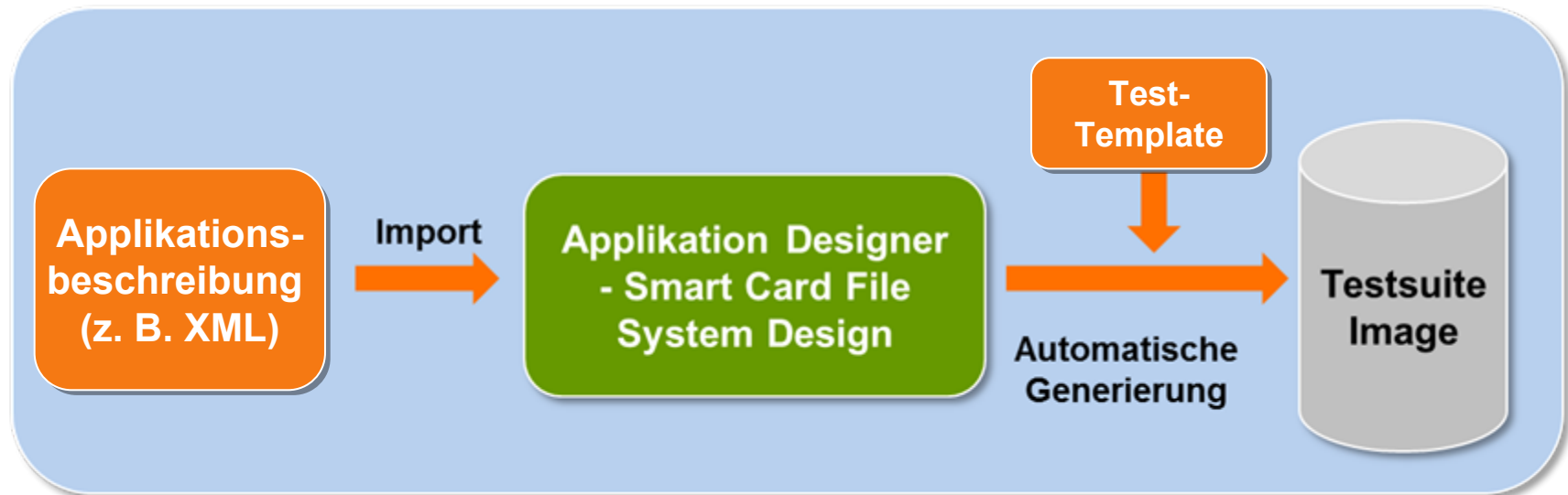
Komponenten der Eclipse RCP Plattform



Plugin-Konzept einer beispielhaften Testumgebung

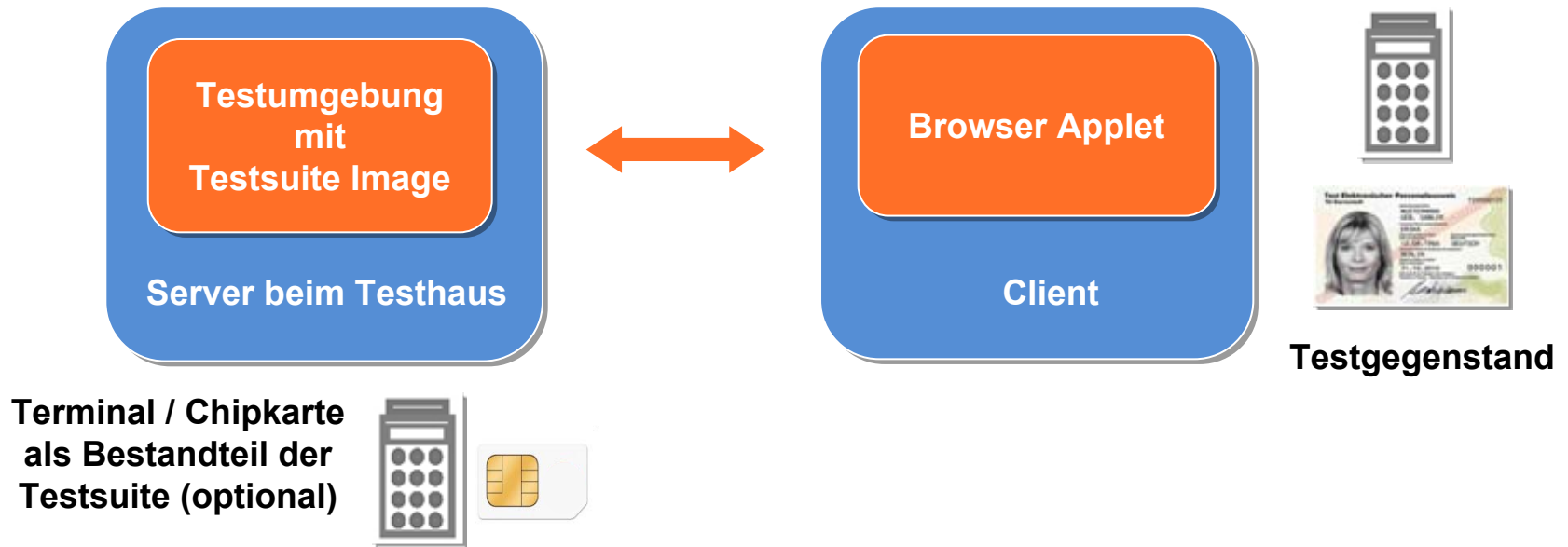


Chipkartenapplikationstest





Konzept Remote Testing



Spezifikationsgetriebene Testfallgenerierung

Festlegung in der Spezifikationsphase

- Testverfahren
- Testkriterien

Umsetzung

- Einführung einer hypothetischen Programmiersprache (Pseudosprache)
- es wird auf existierende Module zugegriffen

Vorteile

- Implementierungsphase entfällt
- keine Programmierkenntnisse erforderlich

Nachteil

- Flexibilität geht verloren



Modellbasiertes Testen

Testprozess wird durch Einsatz von Modellen systematisiert und automatisiert

aus Testmodellen werden Testeingaben, erwartete Testergebnisse und Testscripte generiert

Vorteile

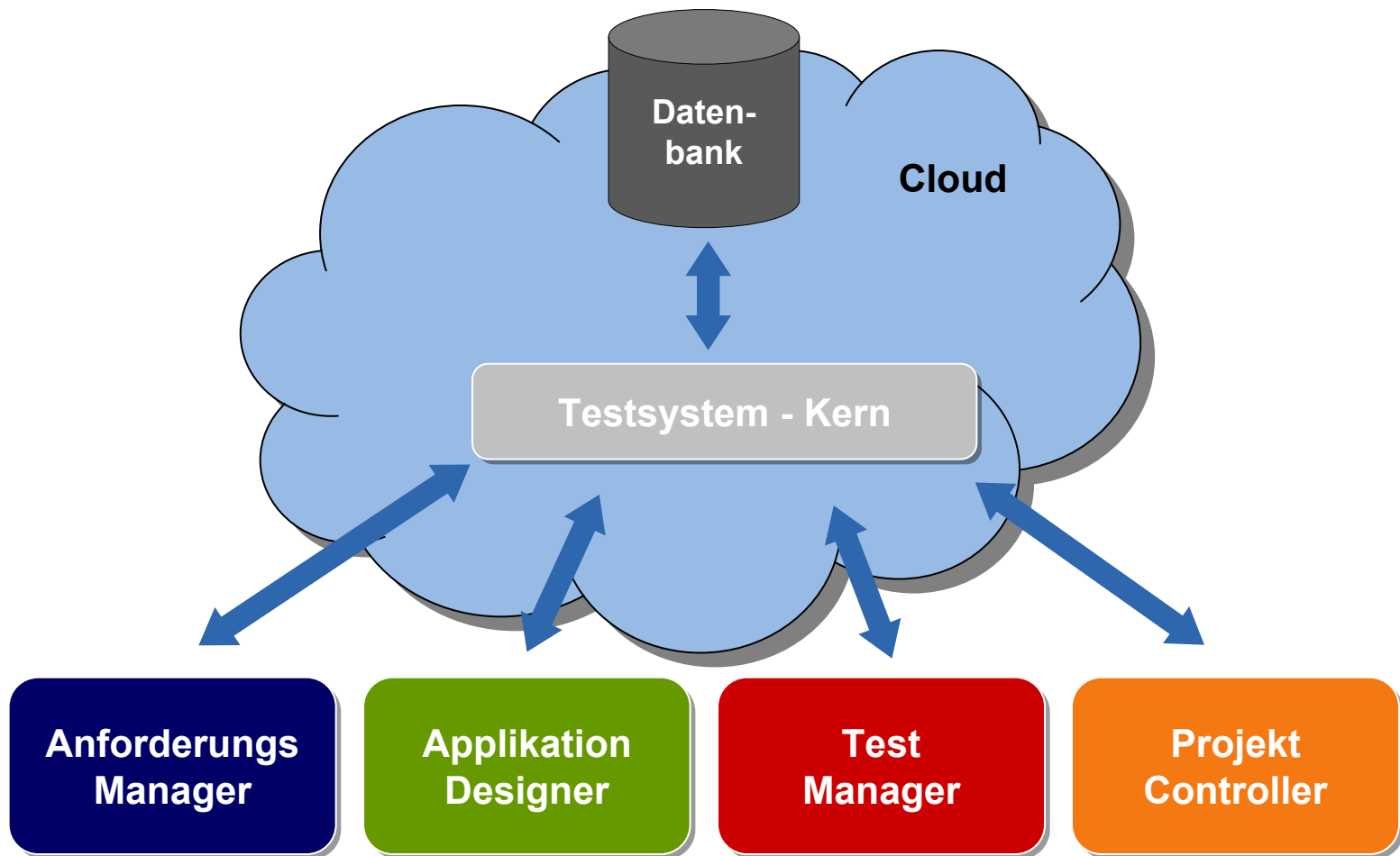
- bei Anforderungsänderungen müssen nur die Testmodelle angepasst werden
- Implementierungsphase entfällt

Nachteil

- hohe Anfangsinvestition (Werkzeuge, Schulung)



Verteiltes Testen in der Cloud





Fazit:

- **keine 100 % Automatisierung des Testzyklus zur Zeit möglich**

Aber durch:

- **Rollenbasierte Testumgebung**
- **Plug-In Konzept (Baukastenprinzip)**
- **Objektorientierte Programmiersprache**
- **Integration von externen Schnittstellen (Interfaces)**

Teilautomatisierungen und Harmonisierung möglich

=> Effektives, flexibles und anwenderfreundliches Testen

Vielen Dank für Ihre Aufmerksamkeit!

Thomas Freitag

Senior Consultant

achelos GmbH

Telefon: +49 (0) 5251 14212-304

Mobil: +49 (0) 175 2929013

Thomas.Freitag@achelos.de

