software quality lab
improve your quality

Quality Knowledge Letter

www.software-quality-lab.at

Ausgabe 2010 / 2

Erscheint ca. 3-4 x jährlich in elektronischer Form

# Testautomatisierung über System-Schnittstellen

## In dieser Ausgabe:


- ⇨ Systemische Sicht auf Schnittstellen
- ⇨ Testautomatisierung über System-Schnittstellen
- ⇨ Literatur und Infos


## Systemische Sicht auf Schnittstellen

Wenn wir Testautomatisierung machen, dann sind wir immer mit den unterschiedlichsten Schnittstellen (im Englischen Ports oder Interfaces genannt) konfrontiert. Im Folgenden meinen wir ausschließlich Softwareschnittstellen, wenn wir von Schnittstellen sprechen. Und davon gibt es reichlich in komplexen, vernetzten Softwaresystemen (auch *Multisysteme* oder *System of Systems* genannt). Jede Komponente kommuniziert mit den anderen über unterschiedlichste Kanäle und Kommunikations– und Interprozessprotokolle. Vielfach ist die Kommunikation indirekt, so dass eine Komponente den Status anderer Komponenten über Umwege beeinflusst.

Bei der Testautomatisierung sind Schnittstellen gleichermaßen für die Ausführung der konkreten Testschritte als auch für die Verifikation der Testergebnisse wichtig und notwendig. Die reine *Blackboxsicht* wird dabei durch eine *Grayboxsicht* ersetzt; wir bewegen uns also auf einer vorwiegend technischen Ebene und benötigen viel Vorwissen über das *System under Test* und die einzelnen Prozesse, die darin ablaufen.

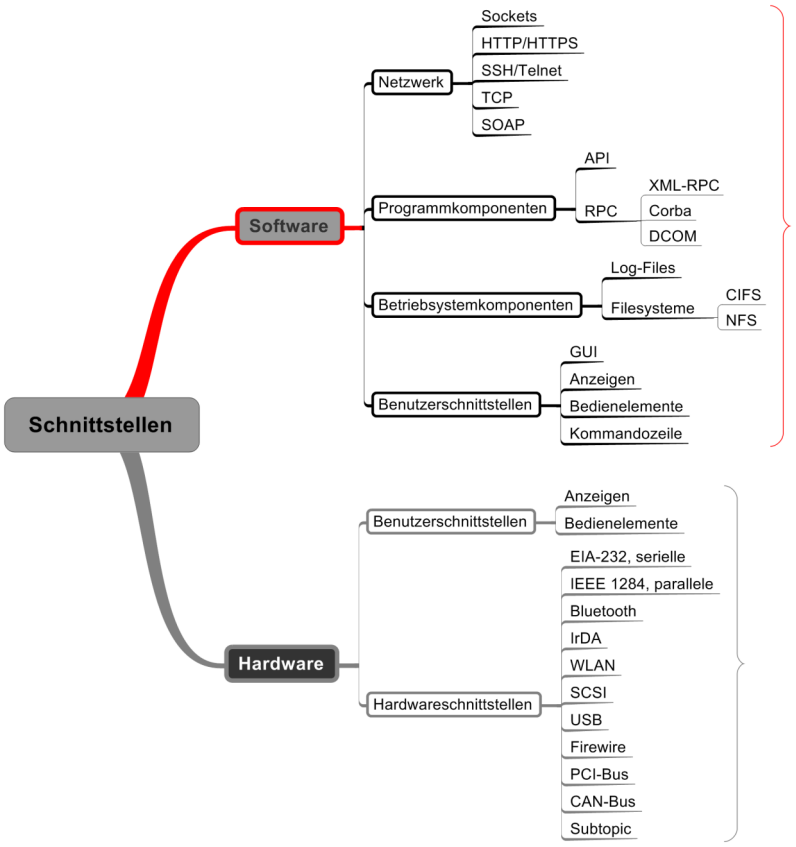


Abbildung 0 – Beispiel für Schnittstellenklassifikation

**Nabelschnur des Lebens der Test-Automatisierung**

Schnittstellen sind notwendig, um Systeme miteinander zu integrieren und Kommunikation zu ermöglichen.

Für die Testautomatisierung sind Schnittstellen zur Kommunikation mit den zu testenden Systemen „lebensnotwendig“. Sowohl die Steuerung der Testautomatisierung als auch das Rückführen und Auswerten der Testergebnisse passiert über Schnittstellen.

Viele Aspekte der Testautomatisierung lassen sich jedoch durch Standard-GUI-Schnittstellen nicht oder nur zum Teil realisieren, sodass die Automatisierung über System-Schnittstellen zielführend ist.

Diese stellen meist auch eine besondere Herausforderung dar, da Standard-Testautomatisierungs-Tools- und -Frameworks oft System-Schnittstellen nur eingeschränkt einbinden können. Es sind fast immer individuelle Anpassungen und Ergänzungen nötig.

Dieser Knowledge-Letter listet einige bekannte System-Schnittstellen auf und zeigt verschiedene Aspekte im Zusammenhang mit der Testautomatisierung.

***Dipl.-Ing. Johannes Bergsmann***
Staatl. befugter und beeideter Ingenieurkonsulent für Informatik

Der Quality-Knowledgeletter ist ein periodisches Informationsmedium von Software Quality Lab und dessen Partnern mit den Schwerpunkten IT-Qualitätsmanagement, Projekt- und Prozess-Management.

Inhalt: fachliche Beiträge und Schwerpunktthemen, Vorstellung neuer Produkte und Leistungen, neue wissenschaftliche Erkenntnisse und andere Fachbeiträge aus unseren Themenbereichen.

Aktuelle Fach- und Forschungsbeiträge sind willkommen. Einsendungen an info@software-quality-lab.at .

**Weitere Infos zu diesem und anderen Themen finden Sie auf http://www.software-quality-lab.at .**

# Testautomatisierung über System-Schnittstellen

von DI Jürgen R. Plasser, Software Quality Lab

**Komplexe Multisysteme sind über eine Vielzahl von unterschiedlichen Komponenten verbunden, die oft nur eingeschränkt von außerhalb der Systemgrenzen erreichbar sind. Glücklicherweise gibt es jedoch APIs, verschiedene Protokolle oder auch Clientapplikationen zur Kommunikation mit diesen Systemkomponenten. Einen Überblick, welche Möglichkeiten zur Testautomatisierung unter Verwendung von bekannten Systemschnittstellen es gibt und was dabei zu beachten ist, gibt folgender Artikel.**

## Allgemeines

Die Anforderung, ein komplexes Multisystem, das nicht explizit für Testen entworfen wurde, einer Testautomatisierung zu unterziehen, ist nicht einfach umzusetzen. Meistens sind eine große Anzahl verschiedener Komponenten nur über bestimmte Protokolle oder APIs ansprechbar, was die Komplexität und Wartbarkeit des Testautomatisierungsframeworks (TAF) beeinflusst und damit die Effektivität der gesamten Testautomatisierung.

Viele Softwareentwicklungen sind von einer kurzen *Time-To-Market-Phase* geprägt, was leider dazu führt, dass weder explizit für Testbarkeit entwickelt wird, noch nichtfunktionale Aspekte der Software, wie etwa Sicherheit, im notwendigen Umfang berücksichtigt werden.

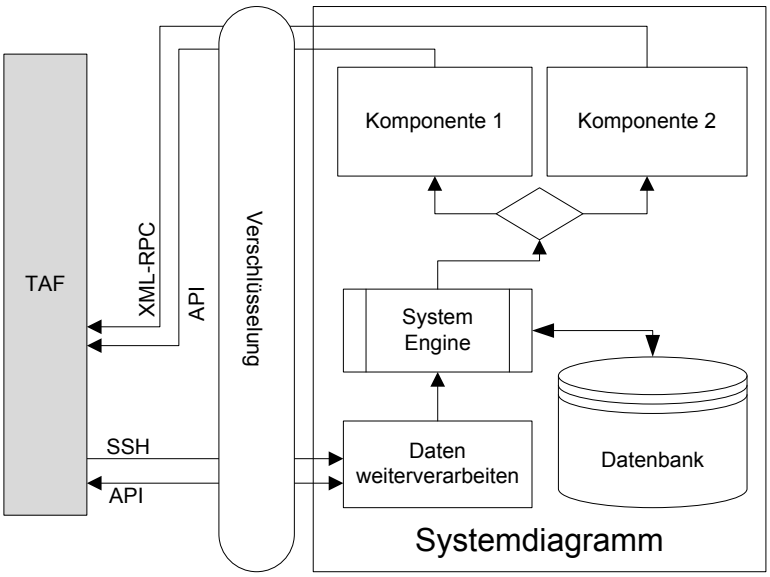


Abbildung 1 – Schnittstellen zum System vom TAF gesehen

## Was wird getestet?

Das zu testende System hat in der Mehrzahl der Fälle eine gewisse Anzahl Komponenten *c*, die über verschiedene Schnittstellen *i* untereinander kommunizieren. Oft sind diese Schnittstellen schlecht dokumentiert oder schwer von außerhalb der Systemgrenzen erreichbar. Zumeist wurde auch zu wenig Wert auf *Design for Testability* gelegt, was sich negativ auf den Aufwand der Testautomatisierung auswirkt.

Die Systemkomponenten sind in der Regel von Zugriffen von außen geschützt und damit nicht direkt für das *Testautomatisierungsframework* (TAF) zugreifbar. Wie kann dann der TAF-Designer Zugriff auf die Systemkomponenten bekommen?

## Zugriff über Schnittstellen

Vielfach bieten die einzelnen Komponenten des Systems Zugriff über verschiedene Protokolle und dadurch definierte Schnittstellen an, um mit der Außenwelt zu kommunizieren. Wie zuverlässig eine Anbindung des TAF über diese Schnittstellen gemacht werden kann, hängt in der Regel stark vom Protokoll ab und wie sehr sich diese Komponente über die Zeit durch die Produktentwicklung verändert.

Als stabile Schnittstellen können meist APIs betrachtet werden, da diese typischerweise schon mit Fokus auf **Wiederverwendbarkeit** und **Wartbarkeit** entworfen wurden. Dazu benötigt der TAF-Entwickler aber eine gut dokumentierte und gewartete Bibliothek in der von ihm gewählten Entwicklungsumgebung, was nicht immer der Fall sein wird.

Wie am Ende dieses Artikels dargestellt wird, gibt es eine Vielzahl anderer Schnittstellen-Protokolle, die den Einsatz von Standard-Libraries ermöglichen und daher auch den meisten Programmiersprachen und Entwicklungsumgebungen zugänglich sind.

## Besondere Herausforderung bei der Testautomatisierung über System-Schnittstellen

Die Kommunikation des Benutzers mit einem System läuft zumeist über eine spezielle Konsolenanwendung mit einer grafischen Benutzeroberfläche und nicht über eine der Systemschnittstellen. Direkte System-Interfaces verwenden meist nur der Systemadministrator oder Supporttechniker, um den Zugriff auf bestimmte Komponenten unabhängig der Restriktionen einer GUI zu erhalten.

# Vollständiger Knowledge Letter Zugang

Wir freuen uns, dass Sie an diesem Thema Interesse haben und den Knowledge Letter von Software Quality Lab bis hierher gelesen haben.

**Dieser Knowledge Letter ist eine Vorschau (gekürzte Version des gesamten Artikels).**

Wenn Sie den ungekürzten Knowledge Letter lesen möchten, registrieren Sie sich bitte unter
**http://www.software-quality-lab.com/download/knowledge-letter/anfrage-knowledge-letter/**

Sie erhalten nach der Registrierung vollen Zugang zu allen bisherigen Knowledge Lettern von Software Quality Lab und erhalten automatisch künftige Knowledge Letter per E-Mail.


**Software Quality Days — Die größte Konferenz zum Thema „Software Qualität“ in Europa!**

Besuchen Sie die Top-Konferenz mit allen Infos rund um Software Qualität.

Beste Qualität der Vorträge und Tutorials sowie eine Mischung aus praktischen und wissenschaftlichen Beiträgen machen die Software Quality Days zum Top-Event.

In den 3 praktischen Tracks werden anwendungsorientierte Vorträge präsentiert. Der wissenschaftliche Track zeigt Beiträge mit hohem Innovationsgrad und praktischer Anwendbarkeit, basierend auf Forschungs-ergebnissen. Im Solution Provider Forum präsentieren Aussteller ihre neuesten Tools mit Praxis-Beispielen.

Nähere Infos unter

**www.software-quality-days.com**