

Antwortfax

+49 (22 03) 91 54-15

Anforderungen von weiteren Informationen:

- PI - kostenloses Produktinformationsseminar (öffentlich)
- PI - Produktinformationsseminar (inhouse, auf Anfrage)
- Informationen zu SQS-TEST®-Schulungen
- unverbindliches Beratungsgespräch
- Interesse an einer Testinstallation
- Sonstiges:

Absender

Name, Vorname

Firma

Funktion

Straße/Postfach

PLZ/Ort

Telefon/Telefax

Datum

Unterschrift

SQS-TEST®

Das System für den fachlichen Software-Test
- Methodenbeschreibung -

SQS-TEST®

das System für den fachlichen Software-Test

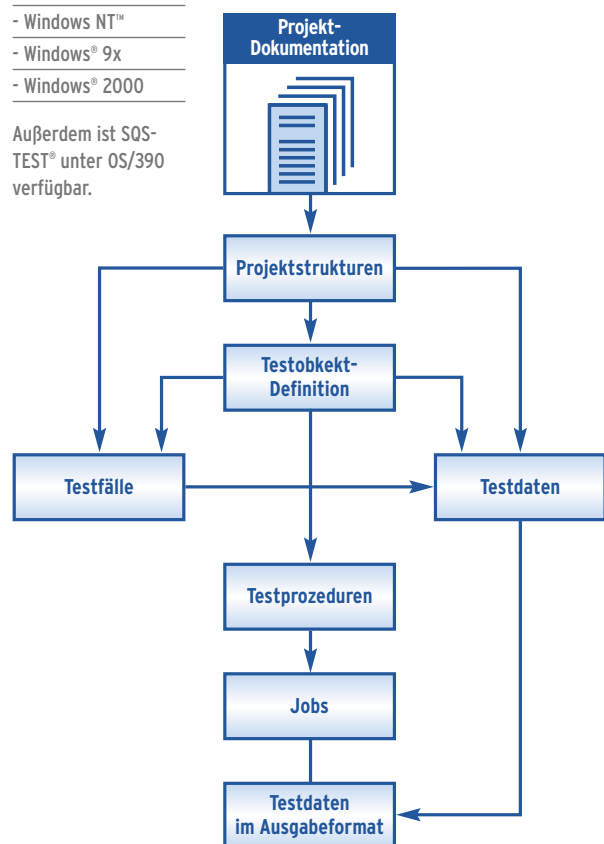
- Methodenbeschreibung -

Professionelles Testen sichert die Qualität, reduziert die Kosten und erhöht den produktiven Anteil der Softwareentwicklung. Leistungsfähige und -geprüfte Produkte sichern Ihnen Wettbewerbsvorteile.

Systemvoraussetzungen für SQS-TEST®
 Server-Plattformen:
 - Aix®
 - HP-UX®
 - SUN/SOLARIS®
 - Windows NT®
 - Windows® 2000

Client-Plattformen:
 - Windows NT®
 - Windows® 9x
 - Windows® 2000

Außerdem ist SQS-TEST® unter OS/390 verfügbar.



Qualität ist (k)eine Kostenfrage

sie **spart Kosten**.

→ Ein Produkt nachzubessern, statt vor dem Einsatz oder Vertrieb zu testen, erhöht die Kosten der Qualitätssicherung um das **Dreifache**. Denn jeder Fehler, der nicht entdeckt wird, ist das erste Glied einer Kette von immer teurer werdenden Folgefehlern.

→ Qualitätssicherung ist eine **Forderung des Marktes**: das Qualitätsbewusstsein wächst, Kunden sind des „Testing by Doing“ überdrüssig.

Wirksame Qualitätssicherung setzt frühzeitig ein.

→ Je eher mit dem Testen begonnen wird, desto optimaler ist das Verhältnis von Aufwand und Nutzen: Die Fehlerrate sinkt, teure Nachbesserungen entfallen.

→ Systematisches Testen vorausgesetzt, können bis zu 70% der Tests vor Fertigstellung der Software durchgeführt werden.

Effektives Testen

■ beginnt frühzeitig, schon bei Vorliegen des Fachkonzeptes und in der Analyse-Phase

■ erfolgt parallel zur Entwicklung
 ■ benutzt regressionsfähige Daten
 ■ testet das Wichtige (zuerst), und das möglichst vollständig.

Testen mit SQS-TEST®

SQS-TEST® – das System für den automatisierten Software-Test

In SQS-TEST® sind fast 20 Jahre Know-how aus Praxis und Forschung eingegangen sowie die Wünsche und Anregungen zahlreicher (professioneller) Anwender.

SQS-TEST® ermöglicht:

■ **Systematische Testvorbereitung** in frühen Stadien der Softwareentwicklung, bevor Module, Programme und Softwarefunktionen fertig gestellt sind.

■ **Automatisierung** bei der Erfassung und der technischen Testausführung

durch Generierungsmechanismen und Testprozeduren.

■ Weitgehend automatisierte Anpassungen, wenn Datenstrukturen und Programmfunktionen geändert werden (Regressionsfähigkeit).

Die Komponenten von SQS-TEST®

SQS-TEST® besteht aus **Komponenten**, die komplett miteinander verzahnt sind:

■ **Strukturverwaltung**: Die für den Test erforderlichen Datenstrukturen (Informationen über Dialoge, Stammdatensätze usw.) werden aus der Projektentwicklung übernommen und bereit gestellt.

■ **Testobjektdefinition**: Die Anwendung wird analysiert und in Testobjekte „zerlegt“. Jedem Testobjekt werden die benötigten Ein- und Ausgabestrukturen zugeordnet. Die Aufteilung in Testobjekte ermöglicht ein Testen nach Prioritäten wie „sicherheitskritisch“, „Standardfunktion“ usw.

■ **SQS-TEST®/Testfallermittlung**: Die Anforderungen an die Testdaten werden entwickelt.

■ **SQS-TEST®/Testdatendefinition**: Die Testdaten werden definiert und im gewünschten Format bereitgestellt.

■ **SQS-TEST®/Testprozessautomatisierung**: Aus Testprozeduren werden Jobs generiert und automatisiert durchgeführt.

SQS-TEST®/Testfallermittlung

Anforderungen an Testdaten entwickeln

Die **eingabeorientierte** Testfallermittlung entwickelt fachliche Anforderungen an Testdaten – in der Regel für einzelne **Funktionen** einer Anwendung. Damit wird die Frage beantwortet: „Welche Eingaben (z.B. in einem Dialog) müssen erfolgen, damit alle kritischen Verarbeitungsprozesse einer Anwendung aufgerufen werden?“ Auf diese Weise können die Tester feststellen, ob das System funktioniert wie vorgesehen.

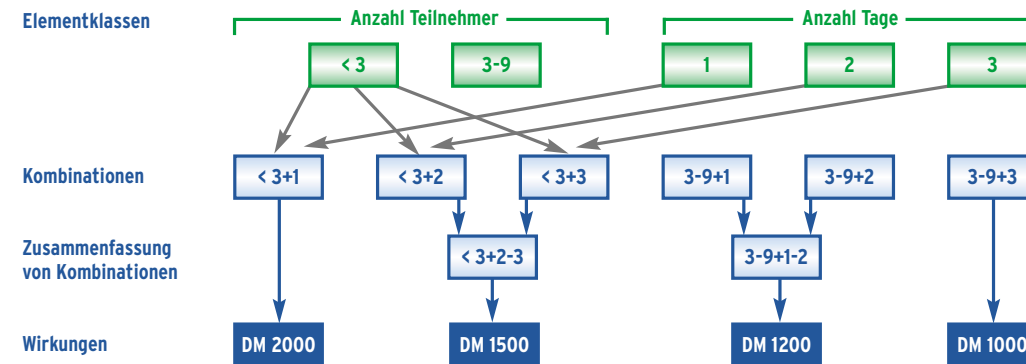
Eingaben und Wirkungen

Verschiedene Eingaben in den Feldern eines Dialoges der zu testenden Anwendung sollen in der Regel auch verschiedene **Wirkungen** zur Folge haben. Ziel eines Tests ist es, zu überprüfen, ob Eingaben die vorgesehenen Wirkungen haben und ob diese Wirkungen eindeutig sind. Es muss einen Unterschied machen, ob „J“ oder „N“ in einem Bestätigungsdialog eingegeben wird.

Gruppierung der möglichen Eingaben durch Äquivalenzklassenanalyse

Da nicht alle überhaupt denkbaren Eingaben durchprobiert werden können, müssen sie reduziert werden. Überschaubar ist die Anzahl der möglichen Wirkungen. Daher gilt bei der Analyse einer Datenstruktur (z.B. eines Dialogs) folgendes Verfahren: Für jedes Datenelement (Eingabefeld) werden die möglichen Eingaben in Gruppen aufgeteilt, die jeweils die gleiche (äquivalente) Wirkung haben – sogenannte **Äquivalenz- oder Elementklassen**. Alle Eingaben mit gleicher Wirkung bilden eine Elementklasse.

Die Eingaben "j", "J", "y", "Y" bilden eine Klasse mit derselben Wirkung, um einen Dialog positiv zu bestätigen, die Eingaben "n" und "N" eine zweite (Wirkung: Abbruch der Bearbeitung); alle anderen Eingaben (z.B. "x" oder "5") wären fehlerhafte Eingaben (Wirkung: Fehlermeldung).



Die Elementklassen dürfen sich nicht überschneiden und enthalten insgesamt alle überhaupt möglichen Eingaben. Aufgabe der späteren Testdatendefinition wird es sein, aus diesen gleichwertigen Eingaben einer Elementklasse die für den Test effektivsten auszuwählen.

In Fällen mit einem vorgegebenen Bereich werden häufig die obersten und untersten Werte (Grenzwerte) und ein Durchschnittswert ausgewählt. In einem Bestätigungsfeld könnten z.B. die Eingaben "j" und "n" sowie "x" verwendet werden.

Kombinationen von Eingaben

In vielen Fällen haben erst mehrere Eingaben gemeinsam eine eindeutige Wirkung.

Z.B. könnte bei einer Seminarbuchung die Höhe der gewährten Rabatte (Wirkungen) abhängig sein von der Anzahl der gemeldeten Teilnehmer (z.B. <3; 3-9) und der Anzahl der gebuchten Tage (1, 2 oder 3) (siehe Grafik).

Für diese Elemente (Felder) werden sogenannte **Abhängigkeiten** gebildet. Dann werden von SQS-TEST® automatisch alle möglichen **Kombinationen** erzeugt, d.h. jede Elementklasse eines Elementes wird mit allen Elementklassen der anderen Elemente verknüpft.

Beim obigen Beispiel werden 2 (Elementklassen der Teilnehmerzahl) x 3 (Anzahl der Tage) Kombinationen erzeugt (siehe Grafik).

Die Kombinationen werden – anhand einer übersichtlichen tabellarischen Darstellung – schrittweise analysiert, bis die Wirkungen vollständig und eindeutig den jeweiligen Eingabekom-

binationen zugeordnet sind. Haben verschiedene Kombinationen dieselbe Wirkung, können sie zusammengefasst werden; so wird durch stetige Reduktion die Übersichtlichkeit erhalten.

Testfälle generieren

Den letzten Schritt stellt die Erzeugung von **Testfällen** dar. Ein Testfall ist eine vollständige, zusammenhängende Folge von möglichen Eingaben (eine sog. **Eingabesituation**) im Zusammenhang mit den erwarteten Wirkungen. In einem Testfall ist für jeweils ein Element (Feld) eine Elementklasse abgedeckt; jede Elementklasse muss in mindestens einem Testfall auftauchen. Bei der Generierung zielt SQS-TEST® darauf, alle notwendigen Eingaben mit möglichst wenig Testfällen abzudecken.

Ergebnisse

Die ermittelten Testfälle bilden die Grundlage für die Erzeugung von Testdaten, bei der für jede Elementklasse mindestens eine der möglichen Eingaben ausgewählt werden muss. Die Ergebnisse der Testfallermittlung können in unterschiedlichen Formaten präsentiert werden, z.B. in Form einer **Testfall-Liste**, die eine übersichtliche Vorlage für die Überprüfung und die Erstellung von Testdaten bietet.

Vorteile der Testfallermittlung mit SQS-TEST®:

- Einheit von Methode und Werkzeug
- Sicherheit durch Vollständigkeitsüberprüfung
- Transparenz durch übersichtliche Bildschirmpräsentation
- schlüssige Auswertungen (Testfallliste) u.a.

SQS-TEST®/Testdatendefinition und -verwaltung

Ausführbare Testdaten erzeugen

Ziel der Testdatendefinition ist die Erzeugung ausführbarer Testdaten wie z.B. vollständiger Eingabedaten für einen Dialog oder Stammdatensätze.

Mit SQS-TEST® gibt es **drei Möglichkeiten**, Testdaten zur Verfügung zu stellen:

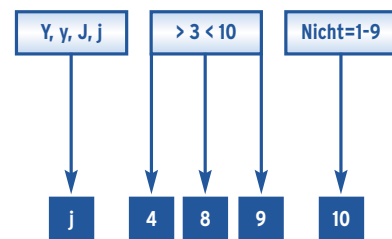
- sie können importiert werden
- sie können aus den mit der Testfallermittlung zusammengestellten Testfällen vorgeneriert und vervollständigt werden
- sie können manuell erstellt werden.

Allgemeine Anforderungen an Testdaten sind:

- es sollten nur so viel wie nötig und so wenig wie möglich erzeugt werden,
- sie sollten möglichst vollständig sein, d.h. alle Anforderungen an die Eingaben (Testfälle) in allen Datenstrukturen abdecken.

Um einen einzelnen Testdatensatz zu erstellen, muss für jedes Element einer Datenstruktur ein **Wert erzeugt** werden, also z.B. für alle Eingabefelder eines Eingabedialogs und die Datenfelder einer Stammdatendatei.

Testfälle:
Mögliche Eingaben



Testdaten:
gewählte Eingaben

Welche Werte für jedes Element erzeugt werden müssen, ergibt sich aus vorher definierten Anforderungen an die Testdaten (Testfälle).

Wenn in einem Datenfeld die Eingaben „y“, „Y“, „j“ und „J“ die selbe Wirkung haben sollen, so muss mindestens eine dieser Eingaben erzeugt werden, um zu überprüfen, ob die Wirkung eintritt. Bei einer möglichen Eingabe von „0 - 9“ kann es z.B. sinnvoll sein, die Grenzwerte „0“ und „9“ sowie einen dazwischen liegenden Wert zu wählen. Für einen Fehlertest können auch Falscheingaben (z.B. „x“ oder „-6“) verwendet werden.

Um den manuellen Aufwand bei der Erfassung der Werte zu reduzieren, stehen in SQS-TEST® eine Reihe von Möglichkeiten zur Verfügung, Werte zuzuweisen.

Globale Wertzuweisung

Bestimmte Daten können in allen Testdatensätzen gleich sein, z.B. Namen, Ortsangaben usw. Für solche Elemente kann der Inhalt einmalig global vorbelegt (und bei Bedarf lokal überschrieben) werden.

Wertzuweisung durch Referenzen

Elemente können durch Referenzierung die Werte anderer Elemente übernehmen.

Auf diese Weise können z.B. Stammdaten, etwa Angaben zu einer Person, konsistent zu Dialogeingaben abgelegt werden; dies ist z.B. erforderlich, wenn in einem Test Eingaben simuliert werden sollen, die auf bereits bestehende Stammdaten zugreifen.

Wertzuweisung durch Variablen

Werte können über globale, explizit definierte Variablen zugewiesen werden. Wird z.B. ein Index verwendet, kann ein Element (z.B. ein Zählerfeld) in den Testdatensätzen nacheinander veränderte Werte erhalten.

Auf diese Weise kann z.B. eine Kennziffer für fortlaufende Datensätze, etwa bei einer Teilnehmerliste, erzeugt werden.

Wertzuweisung durch Ausdrücke

Ausdrücke können Werte, Referenzen und Variablen enthalten und eine

Vielfalt mathematischer Berechnungen umfassen.

So können z.B. Werte in Ausgabefeldern erzeugt werden, die aus den Werten verschiedener Eingabefelder zu berechnen sind.

Direkte, lokale Wertzuweisung

Bei den kritischen Eingabelementen werden in der Regel die Werte direkt und lokal eingetragen.

Eine manuelle Zuweisung für einen einzelnen Fall (eine sog. Testdatenkombination) überschreibt ggf. eine globale Wertzuweisung.

Nach der Zuweisung der Wertvorgaben werden den Testfällen **Testdatenkombinationen** zugeordnet.

Eine Testdatenkombination kann aus einem oder mehreren Testdatensätzen für Strukturen bestehen; die erforderlichen Eingaben werden so kombiniert, dass die Anzahl der Testdaten(sätze) möglichst gering gehalten wird. Die zu einer Testdatenkombination gehörenden Datensätze können zusammenhängend bearbeitet werden.

Wenn Testdaten für die Testausführung bereitzustellen sind, werden die Testdatenkombinationen **Testläufe** zugeordnet. Danach werden die Testdaten physisch im gewünschten Format bereit gestellt.

Um die Eingabe der Testdaten auf der Dialog-Oberfläche automatisch auszuführen, können sie an ein sog. **Capture/Replay-Tool** übergeben werden, das die Eingaben in der Anwendung simuliert. Dies geschieht über von SQS-TEST® aus gesteuerte Testprozeduren.

SQS-TEST®/Testdatendefinition und -verwaltung GUI

Testdaten für Windows-Anwendungen erzeugen

Das Standardverfahren

Um Testdaten für Anwendungen mit grafischer Benutzeroberfläche (Graphic User Interface, GUI) zu erzeugen, werden u.a. Mitschnitt-Werkzeuge eingesetzt, sog. **Capture/Replay-Tools**. Dies ist vor allem bei Windows®-Anwendungen der Fall. Eingaben im zu testenden Dialog werden mit der Capture-Komponente protokolliert und in Skripten abgelegt; diese müssten ohne SQS-TEST® aufwandsintensiv manuell nachbearbeitet werden, ehe sie mit der Replay-Komponente wieder abgespielt werden können. Die so gewonnenen Skripte sind nur bedingt regressionsfähig: Bei Änderungen des Dialoges muss häufig der gesamte Prozess wiederholt werden.

Die Erweiterung der Testdatendefinition mit SQS-TEST® für grafische Benutzeroberflächen

Das Ziel der SQS-TEST®/Testdatendefinition und -verwaltung GUI ist die Erzeugung problemlos ablauffähiger und **regressionsfähiger Skripte für Capture/Replay-Tools**.

Die Testdaten werden zu diesem Zweck **direkt in SQS-TEST® erfasst**. Bei Änderungen der Dialoge der zu testenden Programme ist keine Neufassung erforderlich, da die Testdaten lediglich mit Hilfe der in SQS-TEST® integrierten Abgleichmechanismen angepasst werden müssen.

Zusätzlich erforderliche Informationen

Um in SQS-TEST® Testdaten für GUI-Oberflächen in Form von Capture/Replay-Skripten zu generieren, sind mehr Informationen als die reinen Datenstrukturen der Anwendung erforderlich. Zusätzlich werden

Ressource-Informationen insbesondere über die vom Windows®-System verwendeten Controls benötigt (d.h. die Objekte zur Konstruktion der Oberfläche wie Buttons, Listboxen, Editfelder usw.). Diese werden von SQS-TEST® zusammen mit den Datenstrukturen aus der Entwicklungsumgebung der zu testenden Anwendung importiert; oder sie werden aus der laufenden Anwendung selbst entnommen und in einer eigenen Ressource-Datenbank abgelegt.

Erfassung von Testdaten und Funktionsanweisungen

Um Testdaten für eine GUI-Anwendung mit einem Capture/Replay-Tool ausführen zu können, werden nicht nur fachliche Daten erfasst, sondern zusätzlich die aufeinander aufbauenden **Funktionsanweisungen für jedes Control**. Dies erfolgt über Sprachkonstrukte des Capture/Replay-Tools, die zusätzlich zu den Testdaten erfasst werden.

Die Controls können in der Regel **direkt angesteuert** werden, um z.B. eine Checkbox direkt zu selektieren oder ein Editfeld zu initialisieren, bevor die fachliche Eingabe erfolgt. Als alternative Lösung ist ein Verfahren zur Erzeugung reiner **Tastatureingabesequenzen** möglich.

Simulation der Benutzeroberfläche

Mit Hilfe der Ressource-Informationen ist es möglich, die **Originaldialoge** der zu testenden Anwendung zu **simulieren**. Die **Testdaten und Control-Aktionen** können in dieser Simulation wie in der originalen Anwendung erfasst werden.

Diese Aktionen werden in Befehlen

einer „Instrumentierungssprache“ abgelegt, die bei der Generierung des ablauffähigen Skriptes in die entsprechende Anweisung des Capture/Replay-Tools umgesetzt wird. Diese Befehle können auch manuell eingegeben werden: in einer parallel zum simulierten Dialog angezeigten Tabelle.

Um umfangreiche und wiederkehrende Eingaben zu vereinfachen, können bei der Erfassung der Testdaten **Makros** (parametrisierbare Textbausteine) eingesetzt werden.

Skripterstellung

Aus den erfassten Daten können **Skripte** für nahezu jedes beliebige Mitschnittwerkzeug generiert werden.

Die SQS-TEST®/Testdatendefinition und -verwaltung GUI ist im Lieferumfang von SQS-TEST® enthalten.

Die Erweiterung des Strukturimports steht für folgende Formate zur Verfügung:

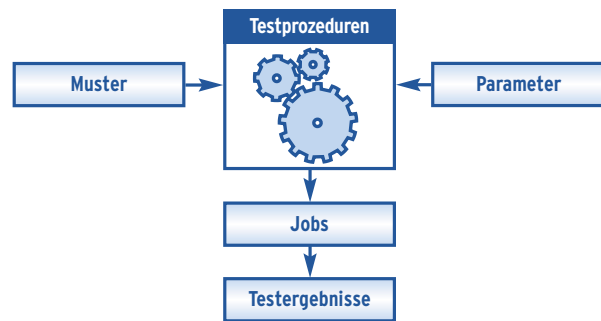
- Microsoft Visual C++ (ab Version 2.0) und Borland® C++ (ab Version 4.5)
- Microsoft Visual Basic® Form-Dateien (FRM) (Version 4.0 und 5.0).
- WinRunner® GUI-Mapping Dateien.
- Erweitertes neutrales Strukturimportformat. Über dieses Format können auf Wunsch weitere Importformate bereitgestellt werden.
- Falls keine Sourcen zur Verfügung stehen, ist eine Extraktion von Struktur- und Ressource-Informationen aus laufenden Windows-Anwendungen möglich.
- Zusätzliche Importschnittstellen in Vorbereitung oder auf Anforderung.

Im Lieferumfang enthalten sind die Anbindungen für folgende Capture/Replay-Werkzeuge (benutzerspezifisch erweiterbar):

- Mercury WinRunner® ab der Version WinRunner 4.03 (deutsch)
- Rational Robot® ab der Version 6.0.0.294
- Rational Visual Test® ab Version 4.0
- Zusätzliche C/R-Tool-Anbindungen in Vorbereitung und auf Anforderung.

SQS-TEST®/Testprozessautomatisierung

Die Testdurchführung organisieren



SQS-TEST®/Testprozessautomatisierung dient zur Steuerung der Testvorbereitung und -ausführung sowie zur Archivierung und Verwaltung der Testergebnisse. Diese Testorganisation erfolgt durch die Ausführung ablaufspezifischer Testskripte (**Jobs**), die aus **Testprozeduren** generiert werden.

SQS-TEST®/Testautomatisierung kann als **Basis für ein individuelles Testsystem** dienen, in dem unter einer einheitlichen Oberfläche unterschiedliche Tools integriert werden. Der Tester kann Standard-Abläufe individuell zusammenstellen und anpassen. Mit Hilfe der Testprozeduren sind systematisch aufgebaute, komplexe **Test szenarien** organisierbar, deren einzelne Stufen durch aktuell generierte Jobs ausgeführt werden.

Eine Testprozedur steuert einen **Testlauf** zum Testen eines Testobjekts in seinem Gesamt Ablauf, z.B. durch:

- Bereitstellen von Testdateien mit Eingaben
- Laden der Testdateien in die Testumgebung
- Starten des zu testenden Programms
- Start eines Capture/Replay-Tools mit der Testdatei für Dialogeingaben
- Speichern und Archivieren der Testergebnisse
- Bereitstellen von Sollergebnisdateien

- Vergleichen der Testergebnis- und zugehörigen Sollergebnisdateien
 - Speichern und Archivieren der Vergleichsergebnisse
 - Beenden aller Jobs und Programme
 - Löschen aller temporären Dateien.
- Um diese einzelnen Aktionen übersichtlich und flexibel zu strukturieren, besteht eine Testprozedur aus einer hierarchisch gegliederten Abfolge von Aufgaben, Aufträgen und Schritten. Eine **Aufgabe** definiert in der Regel ein Teilziel des Testablaufs, z.B. Testvorbereitung, Testausführung, Testarchivierung. Innerhalb einer Aufgabe können mehrere **Aufträge** definiert werden. Sie beschreiben die nacheinander zu erfolgenden Aktionen, die zur Erledigung dieser Aufgabe notwendig sind, z.B. Laden, Ausführen, Sichern der Testdaten. Ein Auftrag besteht wiederum aus einem oder mehreren **Schritten**. In einem einzelnen Schritt werden über die Einbindung von Mustern konkrete, ausführbare Anweisungen definiert.

Diese **Muster** enthalten wiederholt einsetzbare, standardisierte Arbeitsanweisungen (Skript-Templates). Diese sind in der auf dem Zielrechner ausführbaren Skriptsprache geschrieben. Eine individuelle Anpassung erfolgt über **Parameter**, indem Werte an die in den Mustern verwendeten Variablen zugewiesen werden. Diese können in eigenen Dokumenten abgelegt sein, z.B. in Form einer Wertetabelle für Variablen. **Variablen** können nicht nur in der Prozedur eine Wertzuweisung erhalten, sondern alternativ als Eingabevariablen definiert werden; der Wert wird in diesem Fall unmittelbar vor Ausführung eines Testskriptes eingegeben. Die

Steueranweisungen in der Prozedur zur Strukturierung des Testablaufs werden in einer höheren, zielsystemunabhängigen Sprache definiert. Es sind z.B. bedingte Verarbeitungen (#IF, #FOREACH,...) möglich. Aus den Testprozeduren wird pro Auftrag ein lauffähiges **Testskript (Job)** für das jeweilige Zielsystem erzeugt, mit dessen Hilfe die Testprozesse auf dem Zielrechner automatisch ausgeführt werden. Für den Tester erforderliche Informationen können bei der Ausführung angezeigt werden. Informationen über ausgeführte und in der Ausführung befindliche Jobs können über die **Jobkontrolle** eingesehen werden.

SQS-TEST®/Distributed Test Environment - Tests in heterogenen Umgebungen steuern

SQS-TEST®/Distributed Test Environment (DTE) ist eine Erweiterung von SQS-TEST®/Testprozessautomatisierung. SQS-TEST®/DTE ist ein Werkzeug für den automatisierten **Test verteilter Anwendungen**, auch in heterogenen Client-/Server Systemen. Es ermöglicht **netzwerkweit** eine automatische und koordinierte **Ablaufsteuerung** von Testprozeduren über Teiljobs (sog. Tasks) für die verschiedenen Zielrechner.

Im **Lieferumfang** von SQS-TEST® ist eine Teilfunktionalität enthalten, die eine Ausführung und Kontrolle von Teil-Jobs auf dem SQS-TEST®-Server (lokale Tasks) erlaubt. Die Komponenten, die zur Ausführung von Teil-Jobs auf anderen Rechnern erforderlich sind, können zusätzlich für die verschiedenen Zielsysteme erworben werden.

SQS-TEST®/Verarbeitungsorientierte Testfallermittlung

Funktionen im Zusammenspiel testen

Die verarbeitungsorientierte Testfallermittlung unterstützt den Tester beim Aufbau systematischer Anwendungs- und Integrationstests.

Die eingabeorientierte Testfallermittlung mit SQS-TEST® bezieht sich in der Regel auf einzelne Funktionen einer Anwendung: Die in der Anwendung auftretenden fachlichen Funktionen werden analysiert, für jede Funktion wird separat eine systematische Fallbetrachtung durchgeführt.

Eine wichtige **Ergänzung** des Funktionstests stellen **Anwendungs- und Integrationstests** dar. Die Praxis zeigt, dass bestimmte Fehler erst dann in Erscheinung treten, wenn Funktionen nacheinander ausgeführt werden. Beim separaten Test der einzelnen Funktionen würden sie verborgen bleiben. Der Anwendungstest betrachtet daher **Ketten** von hintereinander auszuführenden Funktionen oder Vorgängen und ergänzen damit den auf einzelne Funktionen bezogenen Test.

Drei Arbeitsschritte

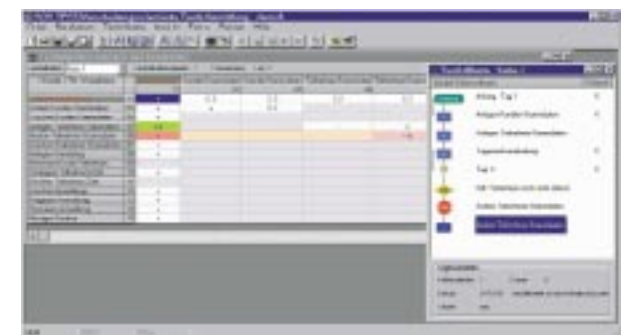
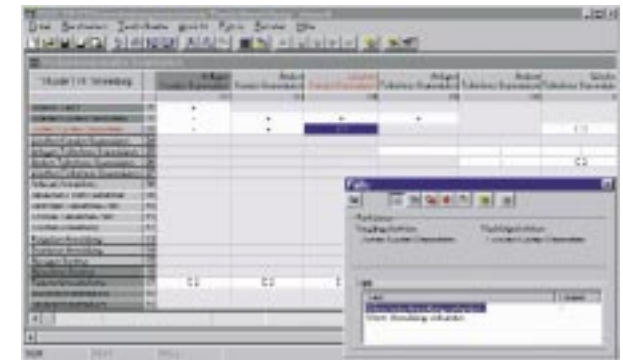
Im **ersten Schritt** werden alle Funktionen einer Anwendung zusammengestellt. Für jede Funktion werden die Datenobjekte, auf die eine Funktion zugreift, mit der Zugriffsart (Lesen, Ändern, Anlegen, Löschen) bestimmt. Ziel ist es, genau jene Funktionen im Zusammenspiel zu testen, die auf gleiche Datenbe-

stände zugreifen. Diese werden zu Testobjekten zusammengefasst und in jeweils einer Tabelle abgebildet. In dieser Tabelle sind die Funktionen so angeordnet, dass jede Funktion mit jeder anderen Funktion einschließ- lich sich selbst kombiniert werden kann. Jedes Funktionspaar stellt eine mögliche Vorgänger-/ Nachfolgerbeziehung dar.

Im **zweiten Arbeitsschritt** erklärt der Tester für jede Vorgänger-/ Nachfolgerbeziehung, ob die Hintereinanderausführung der beiden Funktionen erlaubt, bedingt erlaubt oder nicht erlaubt ist. Dies kann in den Zellen der Tabelle spezifiziert werden. Gleichzeitig legt er fest, ob sich der Zustand des zu testenden Systems durch die Ausführung einer Funktion ändert.

Auf der Basis der fertiggestellten Verbindungsmatrix werden in einem **abschließenden Schritt** alle Vorgänger-/ Nachfolgerbeziehungen in sogenannten **Testfallketten** zusammengestellt. Die Testfallketten bilden die Grundlage für **Testläufe** in der Testausführungsphase.

Zur Synchronisation mehrerer parallel auszuführender Testfallketten, die aus Online-Eingaben und Aufrufen von Batch-Programmen bestehen, können **Szenarien** entworfen werden. Szenarien stellen sicher, dass zeitlich kritische Funktionen nur zu festgelegten Zeitpunkten ausgeführt



SQS-TEST®/Verarbeitungsorientierte Testfallermittlung ist ein separates Werkzeug, das unabhängig vom SQS-TEST® System erworben werden kann.

werden können. Sie bilden ein grobes Zeitgerüst, dem alle Testfallketten genügen müssen.

Die **Oberfläche** der verarbeitungsorientierten Testfallermittlung bietet eine optimale visuelle Unterstützung. Ein integrierter **Testfallkettenmanager** sorgt für die Übersicht über die erzeugten Testfallketten.