

4 Grundlagen von SQS-TEST/Professional New Line

4.1 Einführung

SQS-TEST/Professional New Line (NL) ist ein umfassendes und flexibles Werkzeug für den Test von Softwareanwendungen.

Eine Anwendung (z.B. ein Tool zur Bearbeitung des Kundengeschäfts) wird in der Regel mehrmals in verschiedenen Stadien der Entwicklung getestet.

Testen besagt: es wird überprüft, ob die Software genau die Leistungen fehlerfrei erbringt, die erwartet werden.

4.1.1 Die Softwareanwendung und ihre Funktionen

Eine Softwareanwendung besteht in der Regel aus einer Vielzahl von Funktionen verschiedener Art:

- fachliche Funktionen, z.B. Kunde anlegen, Kunde ändern, Kunde löschen, Konto anlegen, Konto ändern, Konto löschen.
- technische Funktionen, z.B. Anlegen, Ändern und Löschen von Daten unabhängig vom Inhalt.
- Geschäftsprozesse. Dies sind standardisierte Abläufe, bei welchen zumeist fachliche Funktionen in einer bestimmten Reihenfolge hintereinander ausgeführt werden, z.B. enthält der Geschäftsprozess "Kundengeschäft" u.a. die Funktionen Kunde anlegen, Konto anlegen, Kunde dem Konto zuordnen.

Zwischen den Funktionen bestehen bestimmte Abhängigkeiten und Beziehungen:

- Funktionen bilden bestimmte Gruppen, z.B. das Anlegen und Bearbeiten von Kunden und Konten sind Funktionen des Kundengeschäfts.
- Bestimmte Funktionen werden von anderen Funktionen benutzt. Beispiel: Die fachlichen Funktionen "Kunde anlegen" und "Konto anlegen" nutzen die technische Funktion "Datensatz anlegen".

Die Funktionen einer Softwareanwendung und ihre Beziehungen können in SQS-TEST/Professional NL durch den Datenobjekttyp **Systemleistungen** und seine Beziehungen beschrieben werden.

Die zu testende Softwareanwendung wird in ihre Bestandteile (einfache Funktionen und Geschäftsprozesse) zerlegt und durch Systemleistungen beschrieben. Die Beziehungen zwischen den Bestandteilen können ebenfalls erfasst werden:

- die Systemleistungen können hierarchisch strukturiert werden (z.B. zur Gruppierung der Funktionen)
- die Benutzung einer Systemleistung durch eine andere (z.B. die fachliche Funktion "Kunde anlegen" nutzt die technische Funktion "Datensatz anlegen") kann durch Benutzt-Beziehungen dokumentiert werden.

Die Zerlegung der Funktionen und ihre Abbildung durch Systemleistungen sollte unter der Fragestellung erfolgen: welche Einheiten sollen separat und im Zusammenspiel getestet werden?

4.1.2 **Vorgaben für den Test der Software: Testfälle**

Um eine Systemleistung testen zu können, sollten zunächst Vorgaben für den Test spezifiziert werden, d.h. es ist zu fragen:

1. Welche Funktionen sollen getestet werden, d.h. welche bilden separat testbare Einheiten?
2. Was soll getestet werden, d.h. welche Fälle der Ausführung einer Systemleistung (Testfälle) sind zu testen?
3. Welche Vorgaben und Informationen sind erforderlich, damit der Test der einzelnen Testfälle durchgeführt werden kann?

Beispiel:

Um eine Systemleistung durch Testfälle testen zu können, sollte sie eine testbare Einheit bilden. "Kunde bearbeiten" besitzt noch mehrere Unterfunktionen (Kunden anlegen, ändern, löschen), die einen besseren Ansatzpunkt für einen Test bilden, da die möglichen Anwendungsfälle überschaubarer sind.

Die Funktion "Kunde anlegen" besitzt eine Mehrzahl von Anwendungsfällen, z.B.: minderjährigen Privatkunden anlegen, volljährigen Privatkunden anlegen, Firmenkunden anlegen. Jeder dieser Anwendungsfälle kann einen Testfall bilden.

Für jeden Testfall sind bestimmte Vorgaben zu machen:

- **Vorbedingungen:** welche Voraussetzungen müssen gegeben sein, damit die Funktion ausgeführt werden kann? Beispiel: der Kunde ist noch nicht angelegt worden.
- **Eingaben:** welche Eingaben sind möglich bzw. welche dürfen nicht gemacht werden? Beispiel: bei einem minderjährigen Kunden muss beim Test ein entsprechendes Geburtsdatum eingegeben werden.
- **Nachbedingungen:** welche Wirkung soll die Ausführung der Funktion haben? Beispiel: der Kunde wird angelegt bzw. nicht angelegt, wenn eine falsche Eingabe erfolgt ist.
- **Ausgaben:** sollen bestimmte Ausgaben erfolgen? Beispiel: es soll eine Meldung erscheinen, ob der Kunde angelegt wurde oder nicht angelegt werden konnte.

Diese Informationen ermöglichen die Testausführung, d.h. sie liefern Vorgaben für die zu tätigen Eingaben (Testdaten) und die Kriterien (Nachbedingungen, Ausgaben), anhand derer die Ergebnisse überprüft werden können.

Zur Spezifizierung dieser Vorgaben wird in SQS-TEST/Professional NL der Datenobjekttyp **Testfälle** verwendet. Ein Testfall wird speziell für den direkten Test einer bestimmte Systemleistung angelegt.

Um Geschäftsprozesse zu testen, müssen Testfälle für die Funktionen, die dabei hintereinander ausgeführt werden, nacheinander ausgeführt werden. Dabei stellt in der Regel die Nachbedingung einer Funktion zugleich eine Vorbedingung der nächsten. Beispiel: um ein Konto für einen Kunden anlegen zu können, muss vorher der Kunde angelegt sein. In SQS-TEST/Professional NL können Testfälle zu **Testfallketten** zusammengestellt werden. Eine Testfallkette besteht aus Schritten, die Testfälle ausführen.

In einigen Fällen kann es vorkommen, dass Systemleistungen nicht getestet werden sollen, z.B. bei zugekaufter Middleware. Diese Systemleistung wird aber durch die Ausführung derjenigen Funktionen, die sie verwenden, mit ausgeführt. Damit wird sie durch den Test dieser Funktionen mit getestet.

Dies kann in SQS-TEST/Professional NL dokumentiert werden: ein Testfall kann nur eine Systemleistung direkt, aber andere Systemleistungen **indirekt testen**.

Auf diese Weise ist es möglich, eine vollständige Abdeckung von Systemleistungen durch Testfälle zu erzielen.

4.1.3 Anforderungen an die Software

Der Gegenstand des Softwaretestens kann eine Software in allen Stadien der Entwicklung sein. Das erste Stadium ist die Phase der Beschreibung der **Anforderungen** an die Software, d.h. des **Fachkonzeptes**.

Vor der eigentlichen Entwicklung (d.h. der Implementierung der Funktionen) werden Anforderungen an diese definiert und während der Umsetzung modifiziert. Das Fachkonzept beschreibt, welche Leistungen die Software und ihre Bestandteile erbringen soll, z.B. Bearbeitungsfunktionen für Kunden und Kontos bereitzustellen.

Dabei muss folgendes dokumentiert und geprüft werden:

1. Sind die Anforderungen an die Funktionen hinreichend genau beschrieben, nicht widersprüchlich sowie vollständig? Sind die Bedingungen und Ergebnisse der Funktionen vollständig beschrieben, so dass keine zirkuläre Struktur entsteht? Beispiel: um ein Konto anlegen zu können, muss der Kunde bereits angelegt sein. Der Kunde dagegen muss angelegt werden können, ohne dass ein Konto angelegt wird. Anforderungen sind daher selbst – analog zu Systemleistungen – zu testen.
2. Werden die Anforderungen vollständig abgedeckt, d.h. alle beschriebenen Anforderungen durch Systemleistungen implementiert? Beispiel: die Anforderung "Kunde bearbeiten" wird durch mehrere Systemleistungen, nämlich "Kunde anlegen", "Kunde ändern" und "Kunde löschen" implementiert.

In der Komponente "IM Requirements Management" von SQS-TEST/Professional NL können Anforderungen erfasst und gruppiert werden.

Anforderungen können durch Testfälle, ggf. Testfallketten getestet werden.

Die Abdeckung der Anforderungen durch Systemleistungen kann dokumentiert werden.

4.1.4 Projekte organisieren und Testaufträge verteilen

Die bisher verwendeten Datenobjekttypen (Systemleistungen, Anforderungen, Testfälle) bilden die **Produktseite** ab:

- die Anforderungen an die Software
- die Bestandteile der Software (Systemleistungen)
- die Vorgaben für den Test der Software (Testfälle)

Zusätzlich muss der Test des Produktes organisiert und durchgeführt werden, d.h. die Produktseite muss aus **Projektsicht** betrachtet werden, z.B.:

- die zu testenden Systemleistungen werden ausgewählt
- die zugehörigen Testfälle werden Projektmitarbeitern zur Bearbeitung zugewiesen
- die Bearbeitung der Testfälle wird nachgehalten
- der Ist- und Soll-Aufwand für die Bearbeitung wird nachgehalten
- die Systemleistungen werden durch das Testteam freigegeben

4.1.4.1 Arten von Tests

Bei der Organisation der Tests spielen folgende Erwägungen eine Rolle:

1. Ein Test kann funktionsbasiert oder anforderungsbasiert sein. Der Test der Anforderungen und der Systemleistungen kann in SQS-TEST/Professional NL unabhängig voneinander erfolgen.
Bei einem **funktionsbasierten Test** liegt der Schwerpunkt der Betrachtung auf den Systemleistungen und ihrem Test. Je nach gewünschtem Komplexitätsgrad des Tests kann z.B. in diesem Test auch die Abdeckung der Anforderungen durch die Systemleistungen nachgehalten werden.
Bei einem **anforderungsbasierten Test** liegt der Schwerpunkt auf dem Test der Anforderungen. Zusätzlich kann die Implementierung der Anforderungen durch Systemleistungen nachgehalten werden.
2. Der Test kann in mehreren **Teststufen** erfolgen. Die Tests der Funktionen (Entwicklertest, eingabeorientierter Funktionstest) und der Test der Geschäftsprozesse (Business Integration Test oder Integrationstest) werden in der Regel getrennt voneinander durchgeführt. Im Integrationstest werden Fehler aufgedeckt, die erst bei der Hintereinanderausführung der Funktionen auftreten.
Das bedeutet: in den verschiedenen Teststufen werden identische Testfälle mehrfach getestet: im Funktionstest separat, im Integrationstest als Schritt einer Testfallkette.
3. Tests werden in allen Stadien der Entwicklung ausgeführt, d.h. sie müssen parallel zur Entwicklung der Software wiederholt werden können. In **Regressionstests** dienen die Ergebnisse des vorherigen Tests als Basisinformation. Die Behebung dokumentierter Abweichungen muss nachgehalten werden.

4.1.4.2 Testorganisation und Testdurchführung

Bei der Projektorganisation ist zwischen zwei Ebenen zu unterscheiden: der eigentlichen Projektebene oder **organisatorischen Ebene** und der **operativen Ebene**, d.h. der Ebene der Testausführung.

1. Auf der organisatorischen Ebene werden die Systemleistungen bzw. Anforderungen betrachtet. Es wird geplant, welche Systemleistungen bzw. Anforderungen innerhalb eines Teilprojekts oder einer Teststufe getestet werden sollen.
2. Auf der operativen Ebene unterhalb der organisatorischen wird die Durchführung der Tests geplant. Auf der Basis der für die Teststufe bzw. das Teilprojekt ausgewählten Systemleistungen bzw. Anforderungen wird festgelegt, welcher Bearbeiter welche Testfälle bzw. Testfallketten testet, d.h. es werden Testaufträge vergeben und ihre Durchführung wird kontrolliert.
3. Die erforderlichen Aktivitäten und die veranschlagten Aufwände können auf jeder Ebene ermittelt und berechnet werden. Die auf der operativen Ebene geplanten und anfallenden Aufwände werden auf der

organisatorischen Ebene in kumulierter Form benötigt und nachgehalten.

Zur Unterscheidung der Ebenen stehen in SQS-TEST/Professional NL die Datenobjekttypen **Testvorhabenordner** und **Testvorhaben** zur Verfügung:

- Die organisatorische Ebene wird durch Testvorhabenordner strukturiert
- Die operative Ebene wird durch Testvorhaben strukturiert

Grundlage der Projektplanung in SQS-TEST/Professional NL ist die **Definition des Testumfangs**, d.h. die Betrachtung der Datenobjekte der Produktseite aus Projektsicht:

- welche Systemleistungen bzw. Anforderungen sollten im Projekt überhaupt und in den verschiedenen Teststufen und Teilprojekten getestet werden?
- welche Testfälle bzw. Testfallketten sollen von wem getestet werden?

Der Testumfang kann von oben nach unten (Top-Down) verteilt werden:

- Auf der organisatorischen Ebene (Testvorhabenordner) werden die Systemleistungen bzw. Anforderungen selektiert. In einem umfangreichen Projekt können sie auf Unterprojekte verteilt werden, die z.B. von unterschiedlichen Mitarbeitern oder nach unterschiedlichen Prioritäten bearbeitet werden.
- Der auf der organisatorischen Ebene festgelegte Testumfang (ausgewählte Systemleistungen und Anforderungen) wird auf die operative Ebene (Testvorhaben) verteilt. Auf dieser Ebene werden die zugehörigen Testfälle und Testfallketten selektiert und als Testaufträge verteilt.
- Auf jeder Ebene (Testvorhaben und Testvorhabenordner) werden die erforderlichen Aktivitäten definiert und die für jede Aktivität anfallenden Soll-Aufwände ermittelt und die Ist-Aufwände nachgehalten. Auf jeder darüberliegenden Ebene werden die Aufwände kumuliert.