



Testfallermittlung
in der Systemtestpraxis

Harry M. Sneed
harry.sneed@anecon.com



www.anecon.com

Systemtest

auf Basis der Anforderungsanalyse



Von der Idee zum Erfolg!

1 Testen gegen Anforderungen	13 Funktionale Testfälle
2 Gewinnung der Testfälle aus dem Konzept	14 Testfälle für die Grenzwertanalyse
3 Anwendungsfälle und Testfälle	15 Testprozess für ein Websystem
4 Drei Grammartypen deuten auf Testfälle	16 Anforderungsanalyse für den Systemtest
5 Anforderungstext für eine Auftragsbearbeitung	17 Testfall-Datenbank
6 Objekte aus dem Anforderungstext	18 Deutschsprachige Geschäftsregel
7 Aktionen auf Objekte der Auftragsbearbeitung	19 Testfälle aus den Geschäftsregeln
8 Objektzustände in der Auftragsbearbeitung	20 Generierte Testfalltabelle
9 Bedingungen der Auftragsbearbeitung	21 Automatisierte Textanalyse
10 Aktionsorientierte Testfälle	22 Testfälle aus der automatisierten Analyse
11 Zustandsorientierte Testfälle	23 Framework für einen Webapplikationstest
12 Regolorientierte Testfälle	24 Daten aus einem AneCon WebtestProjekt

Qualitätssicherung in der Softwareentwicklung - Seite 2



Anwendungsfälle und Testfälle

ANECON
Von der Idee zum Erfolg!

- ☀ Ein Anwendungsfall hat 1:n Testfälle
- ☀ Es sollten mindestens zwei Testfälle geben

➡ positiver Ausgang
➡ negativer Ausgang

Der Anwendungsfall = **Geld entnehmen**

Ungültige Karte Falsche PIN Nummer Nicht genug Scheine Konto gesperrt Geld auszahlen

Testfälle

Qualitätssicherung in der Softwareentwicklung - Seite 5

Drei Grammartypen

ANECON
deuten auf Testfälle
Von der Idee zum Erfolg!

Aktionen auf Objekte:
Das Kundenkonto wird gesperrt.
Das gewünschte Geld wird ausgezahlt.

Objektzustände:
Das Kundenkonto ist überzogen.
Die Kundenkarte ist abgelaufen.

Bedingungen
Sofern das Konto überzogen ist, wird kein Geld ausgezahlt.
Wenn die Karte ungültig ist, wird sie gesclückt.
Kundenkonten werden gesperrt, wenn sie mehr als 3 Monate überzogen sind.
Im Falle eines Netzausfalls, wird das Geldautomat ausgeschaltet.

Qualitätssicherung in der Softwareentwicklung - Seite 6

Anforderungstext

für eine Auftragsbearbeitung



Von der Idee zum Erfolg!

Das Auftragsbearbeitungssystem bearbeitet die Aufträge, die von den Kunden kommen. Sie werden entweder erfüllt, abgelehnt oder zurückgestellt. Wenn ein Auftrag eintrifft, wird geprüft ob der Kunde bekannt ist und ob seine Bonität in Ordnung ist. Falls der Kunde unbekannt oder seine Bonität nicht in Ordnung ist, wird der Auftrag abgelehnt.

Jeder Kundenauftrag kann bis zu 10 Bestellposten enthalten. Jeder Bestellposten hat die Nummer, der Name und die Bestellmenge des gewünschten Artikels. Erst, wird geprüft ob der gewünschte Artikel auf Lager ist. Wenn nicht, wird diese Bestellung abgelehnt. Als nächstes wird geprüft ob die Artikelmenge auf Lager ausreichend ist, um die Bestellung zu erfüllen. Wenn nicht, wird aus dem Bestellposten eine Rückstellung erzeugt. Falls, der Artikel vorhanden und die Menge ausreichend ist, wird die bestellte Menge abgezogen, die Auslieferung angewiesen und ein Rechnungsposten erstellt.

Immer wenn die Artikelmenge auf Lager reduziert wird, soll geprüft werden, ob die Menge noch über die Mindestmenge ist. Wenn nicht, ist eine Lieferbestellung für den im Moment billigsten Lieferant automatisch zu erzeugen. Ein Lieferauftrag wird aus den Lieferbestellungen eines Lieferanten gebildet und am Ende jeder Woche an die betroffenen Lieferanten versendet.

Die Rechnungsposten sind am Ende jeder Woche zu sammeln und nach Kunde zu ordnen. Jeder Kunde bekommt eine Rechnung für die von ihm in der Woche bestellten Artikel. Der Stückpreis der einzelnen Artikel wird mit der Menge multipliziert, um den Rechnungsbetrag zu ermitteln. Dazu kommt noch der im Moment gültigen Mehrwertsteuersatz. Rechnungen sind innerhalb von 30 Tage zu begleichen.

Qualitätssicherung in der Softwareentwicklung - Seite 7

Objekte aus dem Anforderungstext



Von der Idee zum Erfolg!

1. Auftragsbearbeitungssystem	13. Rechnungsposten
2. Kundenauftrag	14. Mindestmenge
3. Kunde	15. Lieferbestellung
4. Auftragsteller	16. Lieferant
5. Kundenbonität	17. Liefermenge
6. Bestellposten	18. Lieferauftrag
7. Bestellmenge	19. Stückpreis
8. Artikelname	20. Rechnungsbetrag
9. Artikelmenge_auf_Lager	21. Rechnung
10. Rückstellung	22. Mehrwertsteuer
11. Aussendungsanweisung	23. Woche

Qualitätssicherung in der Softwareentwicklung - Seite 8

Aktionen auf Objekten

der Auftragsbearbeitung



Von der Idee zum Erfolg!

<i>1. Aufträge werden bearbeitet</i>	<i>10. Prüfen Menge ausreichend</i>
<i>2. Kunden werden geprüft</i>	<i>11. Lieferbestellung wird erzeugt</i>
<i>3. Kundenbonität wird geprüft</i>	<i>12. Billigste Lieferant wählen</i>
<i>4. Auftrag wird abgelehnt</i>	<i>13. Lieferauftrag wird gebildet</i>
<i>5. Artikel wird geprüft</i>	<i>14. Lieferauftrag wird ausgesendet</i>
<i>6. Artikelmenge wird geprüft</i>	<i>15. Rechnungsposten ordnen</i>
<i>7. Bestellposten wird zurückgestellt</i>	<i>16. Stückpreis wird multipliziert</i>
<i>8. Auslieferung wird angewiesen</i>	<i>17. Rechnungsbetrag wird ermittelt</i>
<i>9. Rechnungsposten wird erstellt</i>	<i>18. Mehrwertsteuer dazu rechnen</i>
<i>10. Bestellpostenen abarbeiten</i>	<i>19. Rechnung zu begleichen</i>

Qualitätssicherung in der Softwareentwicklung - Seite 9

Objektzustände

in der Auftragsbearbeitung



Von der Idee zum Erfolg!

- 1. Kundenaufträge sind erfüllt, abgelehnt oder zurückgestellt*
- 2. Kunde ist bekannt oder unbekannt*
- 3. Kundenbonität ist in Ordnung oder nicht in Ordnung*
- 4. Kundenauftrag enthält bis zu 10 Bestellposten*
- 5. Bestellposten sind erfüllt oder zurückgestellt*
- 6. Artikel ist auf Lager oder nicht auf Lager*
- 7. Bestellmenge ist <,=,> Artikelmenge*
- 8. Rechnungsposten ist erstellt oder nicht erstellt*
- 9. Auslieferung ist angewiesen oder nicht angewiesen*
- 10. Artikelmenge ist <,=,> Mindestmenge*
- 11. Lieferant ist der billigste oder nicht*
- 12. Lieferauftrag ist gebildet & versendet*
- 13. Rechnungsbetrag ist ermittelt*
- 14. Rechnung ist bezahlt oder nicht bezahlt*

Qualitätssicherung in der Softwareentwicklung - Seite 10

Bedingungen
der Auftragsbearbeitung

ANECON

Von der Idee zum Erfolg!

1. Wenn ein Auftrag eintrifft
2. ob der Kunde bekannt ist
3. ob die Kundenbonität in Ordnung ist
4. für jeden Bestellposten
5. ob der bestellte Artikel auf Lager ist
6. ob die Artikelmenge ausreichend ist
7. ob die Artikelmenge über die Mindestmenge ist
8. ob der Lieferant der Billigste ist
9. ob ein Kunde in dieser Woche bestellt hat

Qualitätssicherung in der Softwareentwicklung - Seite 11

Aktionsorientierte Testfälle

ANECON

Von der Idee zum Erfolg!

1. Bearbeite Aufträge	← erteile 1:n Aufträge
2. Prüfe Kunden	← erteile Auftrag vom unbekanntem Kunden
3. Prüfe Kundenbonität	← Kundenbonität in Ordnung
4. Lehne Auftrag ab	← Kundenbonität nicht in Ordnung
5. Prüfe ob Artikel vorhanden	← bestelle nicht vorhandener Artikel
6. Prüfe Artikelmenge	← Artikel vorhanden & Bestellmenge > Artikelmenge
7. Stelle Bestellung zurück	← Artikel vorhanden & Bestellmenge > Artikelmenge
8. Auslieferung anweisen	← Artikel vorhanden & Bestellmenge < Artikelmenge
9. Rechnungsposten erstellen	← Artikel vorhanden & Bestellmenge < Artikelmenge
10. Prüfe ob Menge ausreichend	← Artikelmenge = Mindestmenge
11. Erzeuge Lieferbestellung	← Artikelmenge < Mindestmenge
12. Bilde Lieferauftrag	← erzeuge n Bestellungen von einem Lieferant
13. Rechnungsposten ordnen	← bestelle n Artikel von einem Kunden
14. Stückpreis multiplizieren	← Setze Preise
15. Rechnungsbetrag ermitteln	← erzeuge n Bestellungen von einem Lieferant
• Mehrwertsteuer dazurechnen	← setze Mehrwertsteuersatz
• Rechnung versenden	← Rechnung erstellen
• Rechnung begleichen	← Rechnung erstellen

Qualitätssicherung in der Softwareentwicklung - Seite 12

Zustandsorientierte Testfälle



Von der Idee zum Erfolg!

<ol style="list-style-type: none"> 1. Auftrag ist erteilt 3. Kunden ist nicht vorhanden • Kunden ist vorhanden • Kundenbonität ist nicht ok • Kundenbonität ist ok • Kundenauftrag hat 10 Bestellposten • Bestellposten ist nicht erfüllt • Artikel ist nicht vorhanden • Artikel ist vorhanden • Artikelmenge ist nicht ausreichend • Artikelmenge ist ausreichend • Bestellposten ist erfüllt • Artikelmenge ist zu niedrig 	<ul style="list-style-type: none"> ← erteile einen Auftrag ← erteile Auftrag vom unbekanntem Kunden ← erteile Auftrag vom bekannten Kunden ← erteile Auftrag vom Kunden mit schlechter Bonität ← erteile Auftrag vom Kunden mit guter Bonität ← erteile Auftrag mit 11 Bestellposten ← bestelle nicht vorhandenen Artikel ← bestelle nicht vorhandenen Artikel ← bestelle vorhandenen Artikel ← Bestellmenge > Artikelmenge ← Bestellmenge < Artikelmenge ← Artikel ist vorhanden & Bestellmenge < Artikelmenge ← Artikelmenge < Mindestmenge
--	--

Qualitätssicherung in der Softwareentwicklung - Seite 13

Regelorientierte Testfälle



Von der Idee zum Erfolg!

<ol style="list-style-type: none"> 1. Wenn ein Auftrag eintritt 2. ob der Kunde bekannt ist 3. ob die Kundenbonität in Ordnung ist 4. Für jeden Bestellposten 5. Ob der bestellte Artikel auf Lager ist 6. Ob die Artikelmenge ausreichend ist 7. Ob die Artikelmenge zu niedrig ist 8. Ob der Lieferant der Billigste ist 	<ul style="list-style-type: none"> ← erteile einen Auftrag erteile keinen Auftrag ← Kunde vorhanden in Kundendatenbank Kunde nicht vorhanden ← Kunde mit schlechter Bonität Kunde mit guter Bonität ← erteile Auftrag mit > 1 Bestellposten ← Auftrag.Posten.Art_Nr != E (Article.Art_Nr) Auftrag.Posten.Art_Nr = E (Article.Art_Nr) ← Auftrag.Posten.Bestellmenge > Artikel.Artikelmenge ← Auftrag.Posten.Bestellmenge <= Artikel.Artikelmenge ← Artikel.Artikelmenge < Mindestmenge ← Lieferant.Artikel.Preis = Min (Lieferant.Artikel.Preis)
--	--

Qualitätssicherung in der Softwareentwicklung - Seite 14

Funktionale Testfälle


Von der Idee zum Erfolg!

1.	1. Auftrag:	Auftrag.Kunden_Nr ! e Set (Kunde.Kunden_Nr)	→ unknown customer
2.	2. Auftrag:	Auftrag.Kunden_Nr e Set (Kunde.Kunden_Nr) & Kunde.Bonität != #Adequate	→ Kunde nicht würdig
3.	3. Auftrag:	Auftrag.Kunden_Nr e Set (Kunde.Kunden_Nr) & Kunde.Bonität != #Adequat & Auftrag.Posten_Nr = 0	→ null positions
•	4. Auftrag:	Auftrag.Kunden_Nr e Set (Kunde.Kunden_Nr) & Kunde.Bonität = #Adequat & Auftrag.Posten_Nr > 0 & Auftrag.Posten.Art_Nr !e Set (Artikel.Art_Nr)	→ Artikel nicht auf Lager
•	5. Auftrag:	Auftrag.Kunden_Nr e Set (Kunde.Kunden_Nr) & Kunde.Bonität = #Adequat & Auftrag.Posten_Nr > 0 & Auftrag.Posten.Art_Nr e Set (Artikel.Art_Nr) & Auftrag.Posten.Menge > Artikel.Menge	→ Rückstellung

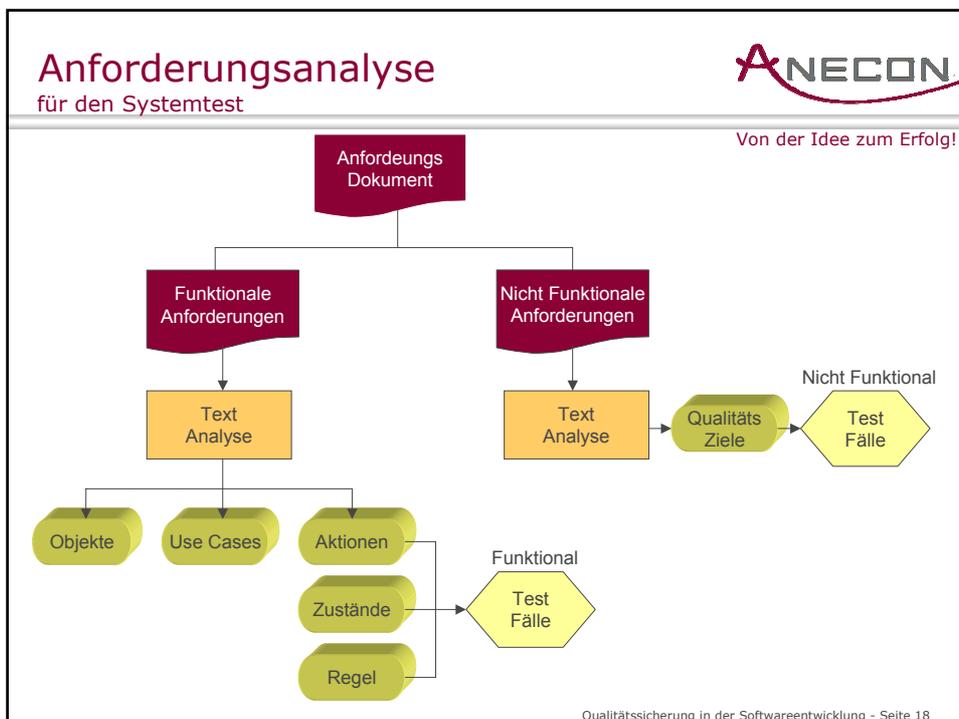
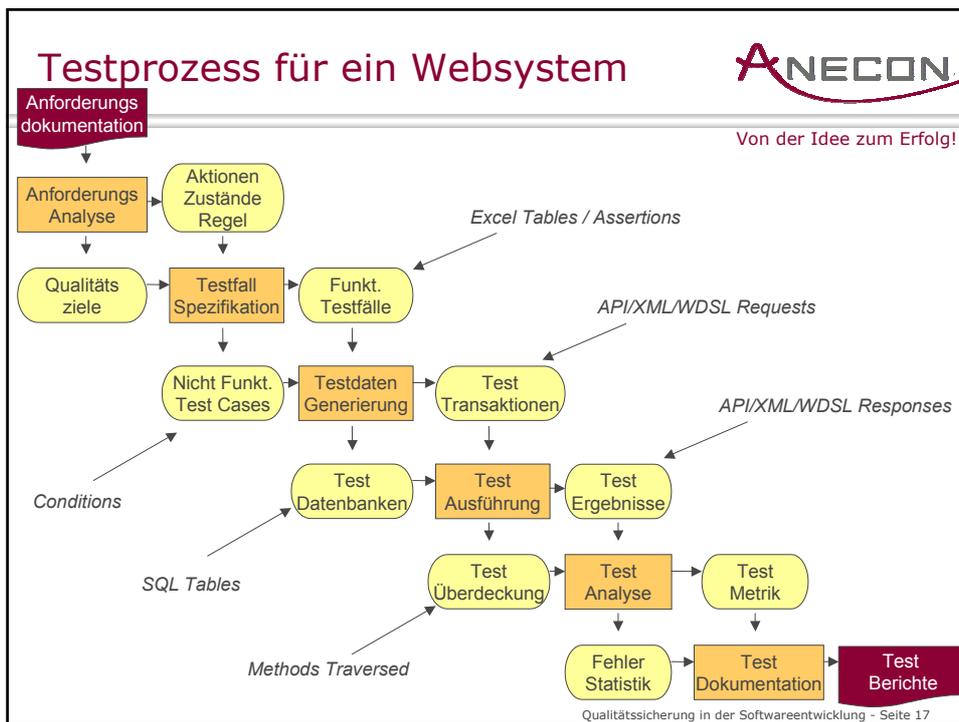
Qualitätssicherung in der Softwareentwicklung - Seite 15

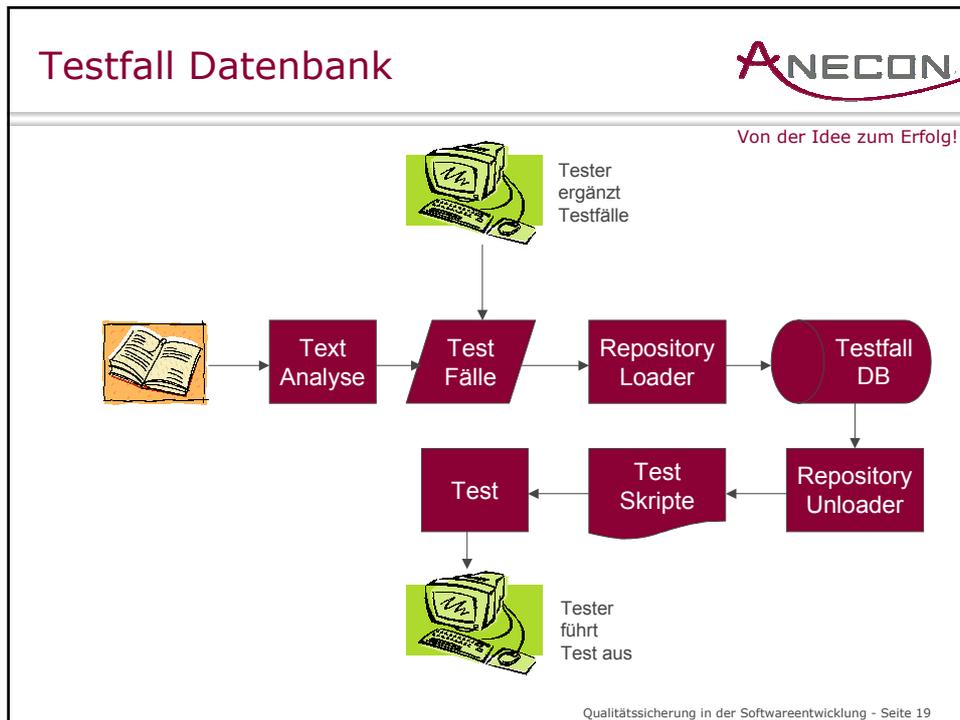
Funktionale Testfälle mit Grenzwertanalyse


Von der Idee zum Erfolg!

1. Auftrag.Posten_Nr = 0, 1, 10, 11
2. Auftrag.Posten.Bestellmenge <, =, > Artikel.Artikelmenge
3. Artikel.Artikelmenge <, =, > Mindestmenge
4. Lieferant.Artikel.Preis <, =, > min (Lieferant.Artikel.Preis)
5. Rechnungsposten <, =, > max (Rechnungsposten)

Qualitätssicherung in der Softwareentwicklung - Seite 16





Deutschsprachige Geschäftsregel

ANECON

Von der Idee zum Erfolg!

Die Zahlungen werden beim Aufbau der Urkunden-Position (nach Bestätigung des Push-Button OK) generiert. Der Zahlungsplan kann solange nachträglich geändert werden, solange keine Zahlungen bearbeitet wurden. Der neue Zahlungsplan wird nach dem Bestätigen des Push-Button OK erstellt.

Das Storno der Geschäftsfälle ist nur durch jeweilige Gegeninstruktionen möglich:

- Storno Eingang -> Ausgang
- Storno Ausgang-> Eingang
- Storno Zahlungsdatum des Inkasso eingegeben -> Zahlungsdatum entfernen
- Storno Zahlungsdatum der Tilgung eingegeben -> Zahlungsdatum entfernen

Modus 'Erfassung'/Bearbeitung: Aufbau/Bearbeiten der Urkunden-Position

über die Such-Pushbutton werden Daten gesucht und ausgew.,hlt.
Die ausgewählten Daten werden im Dialog übernommen. -> siehe Such-Pushbutton

Für eine Schnell-Erfassung können die Daten aus der Ausführung übernommen werden.
Im Sub-Dialog wird die ausgewählte Ausführung für die Dateneübernahme herangezogen.

Die Zahlungen werden generiert, wenn der Dialog mittels Push-Button OK verlassen wird.

Qualitätssicherung in der Softwareentwicklung - Seite 20

Testfälle aus

der deutschsprachigen Geschäftsregel



Von der Idee zum Erfolg!

CMF-TYPE: DIALOG	Date: 17.04.2003
CMF-CONCEPT: GUMKURBE.99C	Requirement Test Specification
CMF-FUNC: DLG-KOBL/Urkunde_Bearbeiten	

TC-NR	TESTTYPE	TESTOBJEKT	=	RULES	PRE CONDITION
102068	Rule=	Ist_FL_gueltig			true JN
102069	Rule=	Ist_VAFI_gueltig			true JN
102070	Rule=	Ist_Ausfg_gueltig			true JN

TC-NR	TESTTYPE	TESTOBJEKT	=	CONDITIONS	PRE CONDITION
102074	solange	nachtraeglich geaendert werden,			true JN
102075	solange	keine Zahlungen bearbeitet wurden.			true JN
102076	wenn	der Dialog mittels Push-Button OK verlassen wird.			true JN
102077	Wenn	Daten fuer die Sub-Dialoge existieren,			true JN
102078	Falls	[\$Param (cSZHLNG_Out.mSTNUM)] = [\$Param (cSZHLNG_In.mSTNUM)]			true JN
102079	wenn	der uebernommene Geschaeftsfall eine andere Art enthaelt			true JN
102080	falls	Pruefungen Warnungen,			true JN
102081	falls	Referenz-Daten im Dialog nicht geprueft wurden,			true JN
102083	Wenn	Stnum uebergeben wurde,			true JN
102084	Wenn	Ausfuehrung uebergeben wurde,			true JN

Qualitätssicherung in der Softwareentwicklung - Seite 21

Generierte Testfalltabelle



Von der Idee zum Erfolg!

```

TYPE;TF_NR ;; CMF ;TEST;AUTO;SYST; :USER ;; DESCRIPTION
CTF ;102068;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;Rule= Ist_FL_gueltig
CTF ;102069;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;Rule= Ist_VAFI_gueltig
CTF ;102070;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;Rule= Ist_Ausfg_gueltig
CTF ;102071;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;Rule= Zahlungsplan_aendern
CTF ;102074;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;solange nachträglich geändert werden,
CTF ;102075;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;solange keine Zahlungen bearbeitet wurden.
CTF ;102076;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;wenn der Dialog mittels Push-Button OK verlassen wird.
CTF ;102077;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;Wenn Daten fuer die Sub-Dialoge existieren,
CTF ;102079;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;wenn der uebernommene Geschaeftsfall eine andere Art enthaelt
CTF ;102080;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;falls Pruefungen Warnungen,
CTF ;102081;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;falls Referenz-Daten im Dialog nicht geprueft wurden,
CTF ;102082;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;Wenn Identifikation uebergeben wurde,
CTF ;102083;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;Wenn Stnum uebergeben wurde,
CTF ;102084;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;Wenn Ausfuehrung uebergeben wurde,
CTF ;102085;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;Wenn Daten schon befuellt sind,
CTF ;102087;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;Wenn Lagerstelle uebergeben wurde,
CTF ;102088;1;0;FUNK;TEST;001 ;URB ;1;WDSSNH; ;Function= DLG-KOBL/Urkunde_Bearbeiten
                    
```

Qualitätssicherung in der Softwareentwicklung - Seite 22

Automatisierte Textanalyse



OBJECTS USE CASES	ACTIONS/STATES/RULES -> PREDICATES	Von der Idee zum Erfolg!
<p>OBJECT: Lebenslage</p> <p>USECASE: "Lebenslage suchen"</p> <p>ACTION: Lebenslage wird angefordert.</p> <p>USECASE: 3.1.1.1 Struktur einer Lebenslage</p> <p>STATE : Lebenslage ist hierarchisch in einem Lebenslagenbaum strukturiert.</p> <p>USECASE: "Verfahrensbeschreibung suchen"</p> <p>RULE : oder es wird ein zu einer Lebenslage gehoerige Verfahrensbeschreibung ueber die Standardnavigation ausgewaehlt.</p> <p>PREDICATE : Lebenslage</p> <p>PREDICATE : Verfahrensbeschreibung</p> <p>PREDICATE : Standardnavigation</p> <p>USECASE: "Alphablock-Suche ausfuehren"</p> <p>RULE : Ob der Landesinhalt fuer eine konkreten Lebenslage / Verfahrensbeschreibung oder die kommunale Erweiterung angezeigt wird, haengt von dem Vorhandensein des entsprechenden Landesinhalt</p> <p>PREDICATE : Landesinhalt</p> <p>PREDICATE : Lebenslage</p> <p>PREDICATE : Verfahrensbeschreibung</p> <p>USECASE: "Formular-/ Onlinedienst-Info suchen"</p> <p>RULE : ob Lebenslagen Informationen zu einem Formular/ Onlinedienst angezeigt werden.</p> <p>PREDICATE : Informationen</p> <p>PREDICATE : Formular/</p> <p>USECASE: "Alphablock-Suche ausfuehren"</p> <p>RULE : welche von den zentral gepflegten Landesinhalten (Lebenslagen / Verfahrenbeschreibungen) oder welche nur regionale Bedeutung haben.</p> <p>PREDICATE : Landesinhalten</p> <p>PREDICATE : Lebenslagen</p> <p>PREDICATE : Verfahrenbeschreibungen</p> <p>RULE : Bei den Lebenslagen koennte der Lebenslagenbaum mit den kommunalen Erweiterungen bei nicht erfolgter Regionalisierung des Nutzers beispielhaft folgende Struktur aufzeigen.</p> <p>PREDICATE : Lebenslagen</p> <p>PREDICATE : Lebenslagenbaum</p> <p>PREDICATE : Erweiterungen</p>		

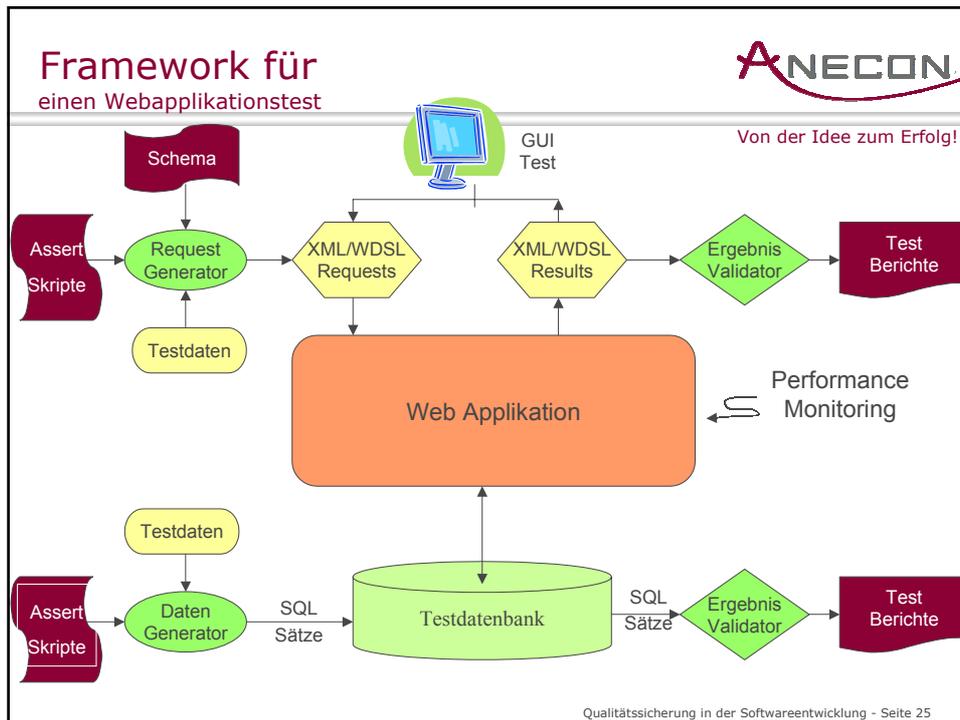
-----Qualitätssicherung in der Softwareentwicklung - Seite 23-----

Testfälle aus der automatisierten Textanalyse



TestCase, Use Case Title	,Relation, TestType, TestCase Description,
TC-0082 ,Behoerde suchen	,perform ,Action ,Verwaltungsstelle regionalisiert anbietet.,
TC-0083 ,Behoerde suchen	,perform ,Action ,Online-Dienste zu verzweigen.,
TC-0084 ,Behoerde suchen	,perform ,Action ,Behoerden angezeigt.
TC-0085 ,Behoerde suchen	,perform ,Action ,Ergebnis ebenfalls direkt angezeigt.
TC-0086 ,Behoerde suchen	,perform ,Action ,Behoerde werden angezeigt.
TC-0087 ,Behoerde suchen	,perform ,Action ,Dienststelle (Behoerde/Abteilung) wird angezeigt.
TC-0088 ,Behoerde suchen	,perform ,Action ,Hierarchien bleiben unberuecksichtigt.
TC-0089 ,Behoerde suchen	,perform ,Action ,Link-Liste angezeigt.
TC-0090 ,Behoerde suchen	,perform ,Action ,Verfuegung stehen.
TC-0091 ,Behoerde suchen	,perform ,Action ,Formularservern gespeichert sein.
TC-0092 ,Behoerde suchen	,perform ,Action ,Dritt-Anbieter realisiert sein.
TC-0093 ,Behoerde suchen	,perform ,Action ,Meta-Informationen abgelegt.
TC-0094 ,Behoerde suchen	,perform ,Action ,Behoerden zugeordnet.
TC-0095 ,Behoerde suchen	,perform ,Action ,Onlinedienst-Info suchen
TC-0096 ,Behoerde suchen	,perform ,Action ,Unterkapiteln naeher beschrieben.
TC-0096 ,Behoerde suchen	,checks ,oder , es wird eine Behoerde ueber die Standardnavigation ausgewaehlt.
TC-0096 ,Behoerde suchen	,checks ,nicht , es wird eine Behoerde ueber die Standardnavigation ausgewaehlt.
TC-0098 ,Behoerde suchen	,checks ,welche , alle Behoerden nach Behoerdengruppen sortiert anzeigt.
TC-0098 ,Behoerde suchen	,checks ,nicht , alle Behoerden nach Behoerdengruppen sortiert anzeigt.
TC-0100 ,Behoerde suchen	,checks ,Ist , einer Behoerdengruppe lediglich eine Behoerde zugeordnet
TC-0100 ,Behoerde suchen	,checks ,nicht , einer Behoerdengruppe lediglich eine Behoerde zugeordnet
TC-0102 ,Behoerde suchen	,checks ,Sollte , mehr als eine Behoerde uebergeordnet
TC-0102 ,Behoerde suchen	,checks ,nicht , mehr als eine Behoerde uebergeordnet
TC-0104 ,Behoerde suchen	,checks ,Sofern , die Regionalisierung des Nutzers nicht bekannt ist
TC-0104 ,Behoerde suchen	,checks ,nicht , die Regionalisierung des Nutzers nicht bekannt ist
TC-0106 ,Behoerde suchen	,checks ,Falls , die Linkliste nur einen Treffer beinhaltet
TC-0106 ,Behoerde suchen	,checks ,nicht , die Linkliste nur einen Treffer beinhaltet
TC-0108 ,Behoerde suchen	,checks ,oder , Online-Diensten.
TC-0108 ,Behoerde suchen	,checks ,nicht , Online-Diensten.
TC-0111 ,Behoerde suchen	,checks ,state ,Dienst sind dies externe Datenquellen
TC-0112 ,Behoerde suchen	,checks ,state ,Anwendungsfaelle sind zu beruecksichtigen:

-----Qualitätssicherung in der Softwareentwicklung - Seite 24-----



Daten aus einem ANECON Websystemtestprojekt

ANECON
Von der Idee zum Erfolg!

- ➔ Portal mit Anbindung eines Formulareservices
- ➔ Ca. 1900 Testfälle wurden spezifiziert und durchgeführt
- ➔ 1390 Testfälle davon für den funktionalen Test der Kernapplikation
- ➔ Aus der automatisierten Textanalyse waren etwa 950 Testfälle abzuleiten
- ➔ Verhältnis Testfälle zu Fehler/Mängel: 1:0,35
Anmerkung: angestrebte Range: 1:0,2 – 1:1,0 (Systemtest)
- ➔ DDP (Defect Detection Percentage) für den Systemtest nach Abnahmetest lag bei etwa 89%
[„Testing is becoming truely effective if DDP > 80%“ – Source: Alistair Gerrard]
- ➔ Kunde war mit den Testergebnissen voll zufrieden

Qualitätssicherung in der Softwareentwicklung - Seite 26



Software Design und Beratung G.m.b.H.
Alser Str. 4/Hof 1 Tel.: +43 1 409 58 90 - 0
A - 1090 Wien Fax: +43 1 409 58 90-998
www.anecon.com office@anecon.com

Vielen Dank für Ihre Aufmerksamkeit!

Harry M. Sneed
harry.sneed@anecon.com



Von der Idee zum Erfolg!

www.anecon.com