

Nicolas Welitzki

Testverfahren zu Business Intelligence Lösungen im Kontext der Qualitätssicherung

Vorgehen, Lösungen und Automatisierung

Welitzki, Nicolas: Testverfahren zu Business Intelligence Lösungen im Kontext der Qualitätssicherung: Vorgehen, Lösungen und Automatisierung. Hamburg, Diplomica Verlag GmbH 2014

Buch-ISBN: 978-3-8428-8846-3

PDF-eBook-ISBN: 978-3-8428-3846-8

Druck/Herstellung: Diplomica® Verlag GmbH, Hamburg, 2014

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Bearbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Die Informationen in diesem Werk wurden mit Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden und die Diplomica Verlag GmbH, die Autoren oder Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für evtl. verbliebene fehlerhafte Angaben und deren Folgen.

Alle Rechte vorbehalten

© Diplomica Verlag GmbH
Hermannstal 119k, 22119 Hamburg
<http://www.diplomica-verlag.de>, Hamburg 2014
Printed in Germany

Inhaltsverzeichnis

Abbildungsverzeichnis.....	IV
Tabellenverzeichnis.....	V
Listings-Verzeichnis	VI
Abkürzungsverzeichnis	VII
1 Einleitung	1
1.1 Aufgabenstellung und Zielsetzung	1
1.2 Vorgehen	1
1.3 Aufbau des Buches.....	2
2 Grundlagen	4
2.1 Grundlegende Begriffe.....	4
2.1.1 Tests und Testverfahren	4
2.1.2 Softwarequalität	4
2.1.3 Business Intelligence	5
2.1.4 Datenaufbereitung.....	5
2.1.5 Data Warehouse	6
2.1.6 Analyse von multidimensionalen Datenräumen	6
2.1.7 BI-Projekte	7
2.2 Analytische Maßnahmen zur Qualitätssicherung.....	8
2.2.1 Testphasen in der Software-Entwicklung	8
2.2.2 Testen in der Software-Entwicklung.....	9
2.2.2.1 Dynamische Testverfahren	9
2.2.2.2 Statische Testverfahren	11
2.2.2.3 Fehlerbasierte Testverfahren	13
2.2.2.4 Erfahrungsbasierte Testverfahren.....	14
2.3 Testen im BI-Umfeld	15
2.3.1 Pre-Deployment Testing nach Kimball	15
2.3.2 Data-Quality Screening nach Kimball.....	15
2.3.3 Design-Review nach Inmon	16
2.3.4 Prävention auf Prozessebene nach Becker, Eberhardt 2009....	16
2.3.5 Testkonzept nach Schröder 2002	17
2.3.6 Testaktivitäten nach Golfarelli, Rizzi 2009	17
2.3.7 Einfache Stichprobentests	18
2.4 BI-Projektphasen	18
2.4.1 Vorgehen	18
2.4.2 Analyse	19
2.4.2.1 Ist-Analyse	19
2.4.2.2 Anforderungsanalyse	19
2.4.2.3 Analyse der Datenquellen	20
2.4.2.4 Erste Fokussierung	20
2.4.3 Konzeption	20
2.4.4 Umsetzung.....	21
2.4.5 Betrieb und Wartung	21

3	Einordnung der Testverfahren.....	22
3.1	Anforderungen an die Testverfahren	22
3.1.1	Anforderungen aus der Sicht der DWH-Architektur.....	22
3.1.2	Anforderungen an ETL-Prozesse aus semantischer Sicht	24
3.1.3	Anforderungen aus Sicht der Datenqualität.....	24
3.2	Abgrenzung der klassischen Testverfahren	25
3.3	Testscenarios mit Werkzeugen wie BI.Quality 2.0	26
3.3.1	Strukturtests	27
3.3.2	Datenvollständigkeits-Tests.....	28
3.3.3	Fremdschlüssel-Tests	28
3.3.4	Unknown-Member-Tests	29
3.3.5	Constraint-Tests	29
3.3.6	Vergleichstests	29
3.3.7	Geeignete Testscenarios	30
3.3.8	Zusammenfassung.....	30
4	Testen im BI-Projekt.....	32
4.1	Testverfahren aus der Software-Entwicklung in BI-Projekten	32
4.1.1	Übersicht	32
4.1.2	Testverfahren in der Projektphase Analyse.....	33
4.1.3	Testverfahren in der Projektphase Konzeption.....	34
4.1.3.1	Design-Review	34
4.1.3.2	Diagramme und Tabellen	34
4.1.4	Testverfahren in der Projektphase Umsetzung	35
4.1.4.1	Dynamische Testverfahren.....	35
4.1.4.2	Statische Testverfahren.....	38
4.1.4.3	Erfahrungsbasierte Testverfahren	42
4.1.5	Testverfahren in der Projektphase Betrieb und Wartung.....	43
4.1.5.1	Integrations- und Bereitstellungstest	43
4.1.5.2	Freigabe- und Benutzertests	43
4.2	Vorgehensmodell für Testverfahren in BI-Projekten	44
4.3	Editor für Unit-Tests in BI-Projekten.....	49
4.3.1	Funktionen des Editors.....	50
4.3.1.1	Hauptprogramm	50
4.3.1.2	Testfunktionen.....	51
4.3.1.3	Verbindungsfunktionen.....	51
4.3.2	Umsetzung	51
4.3.3	Grafische Benutzeroberfläche	52
4.3.4	Neu entwickelte Testscenarios.....	54
4.3.4.1	Schwellwerttest	55
4.3.4.2	Grenzwerttest	55
4.3.4.3	WhiteList-Test	56
4.3.4.4	Plausibilitätstest.....	56
4.3.5	Weiterentwicklung	56
4.3.5.1	Suche nach Tests.....	57
4.3.5.2	Verschiebung von Zeilen/ XML-Tags	57
4.3.5.3	ParameterAuslagerung in Konfigurationsdateien.....	57
4.3.5.4	Hinzufügen von Test-Bausteinen	57

4.3.5.5	Auswahlliste für Referenzen auf Datenbankabfragen	58
4.3.5.6	Import von Verbindungen aus ETL-Paketen	58
4.3.5.7	Generierung der MSBuild-Steuerdatei	58
4.3.5.8	Mehrsprachige Anwendung.....	58
4.3.6	Methodik zum Testen von BI-Projekten mit Unit-Tests	58
4.4	Statische Prüfung von Standards und Konventionen.....	61
4.4.1	Funktionen eines XML-Parsers	62
4.4.1.1	Beispiel 1: Namenskonvention	63
4.4.1.2	Beispiel 2: Schnellauswahl der OLE-DB-Quelle.....	63
4.4.1.3	Beispiel 3: Verwendung von Datenbank-Sichten	64
4.4.1.4	Beispiel 4: Verwendung der „Fast Load Options“	65
4.4.1.5	Beispiel 7: Vermeidung von ressourcenintensiven Komponenten	66
4.4.1.6	Beispiel 8: Lookup-Komponenten-Muster Vermeidung ...	66
4.4.1.7	Beispiel 9: Vermeidung von blockierenden Komponenten	67
4.4.2	Umsetzung der Anforderungen	68
4.4.2.1	Datei öffnen und Komponenten auslesen	69
4.4.2.2	Datenfluss-Komponenten auslesen	69
4.4.2.3	Auswertung der Datenfluss-Komponenten.....	70
4.4.2.4	Auslesen des XML-Tags „property“	70
4.4.2.5	Prüfung zweier XML-Tags.....	71
4.4.2.6	SQL-Kommando-Parser.....	71
4.4.2.7	SQL-Auswertung mit regulärem Ausdruck	72
4.4.3	Weiterentwicklung.....	72
4.4.3.1	Eindeutige Zuordnung der Komponenten über Identifikationsnummern.....	73
4.4.3.2	Programmausführung mit Parameterübergabe ohne GUI.....	73
4.4.3.3	Verwendung von Konfigurationsdateien.....	73
4.4.3.4	Verwendung eines Regelkatalogs.....	73
4.4.3.5	Auswertung anderer XML-Formate	74
4.4.3.6	Einhaltung DB-spezifischer Performance-Regeln	74
4.4.3.7	Performance-Auswertung.....	75
4.4.4	Verwendung im Entwicklungsprozess.....	75
4.5	Erfahrungsbasierte Listen im Entwicklungsprozess	76
4.5.1	Klassifizierung von Fehlern	77
4.5.2	Einfügen von Fehlern in die Liste	79
4.5.3	Verwendung der Liste und Sichten	80
4.5.4	Verwendung im Entwicklungsprozess	81
5	Fazit und Ausblick	82
	Literaturverzeichnis	XI
	Anhang.....	XVII
A	Verwendete Software	XVII
B	Übersetzung ausgewählter SSIS-Komponenten.....	XVII
C	Abrufbare Software aus dem Internet	XVII

Abbildungsverzeichnis

Soweit nicht anders vermerkt, sind die Abbildungen eigene Darstellungen.

Abbildung 1: Data Warehouse Architektur (Vgl. Schröder 2002, S. 4)	23
Abbildung 2: Aufschlüsselung von Daten innerhalb eines Datensatzes (Vgl. Olsen 2003, S. 33)	25
Abbildung 3: SQL-Abfrage und Ergebnis des Strukturtests	28
Abbildung 4: Screenshot vom BI.Quality-Projekt in NUnit	30
Abbildung 5: Übersicht über die übertragenen Testverfahren	32
Abbildung 6: Kontrollflussgraphen mit SSIS-Kontroll- und Datenflüssen	37
Abbildung 7: Beispiel eines Kontrollflussgraphen mit Verzweigung	38
Abbildung 8: Beispiel für falsche Logik beim bedingten Teilen	39
Abbildung 9: Beispiel für die Darstellung von Metadaten	39
Abbildung 10: Beispiel für den Editor zum Auflösen von Spaltenverweisen	40
Abbildung 11: Testarme und testintensive BI-Projekte nach Projektphase	49
Abbildung 12: Das Test-Vorgehensmodell zusammengefasst	49
Abbildung 13: GUI des Editors	52
Abbildung 14: GUI für geöffnete Tests	53
Abbildung 15: Beispiel für XML-Tag AssertValue	53
Abbildung 16: Auswahl der Verbindungen aus dem Projekt	53
Abbildung 17: Screenshot vom Editor nach Programmstart	54
Abbildung 18: Screenshot eines Tests in Bearbeitung	54
Abbildung 19: Schwellwerttest mit SQL-Kommando und Ergebnis	55
Abbildung 20: Variablen setzen mit dem Test-Projekt-Verzeichnis	60
Abbildung 21: Beispiel für einen HTML-Bericht	61
Abbildung 22: Beispiel OLE-DB-Quelle mit Schnellauswahl	63
Abbildung 23: Beispiel OLE-DB-Quelle mit SQL-Kommando	63
Abbildung 24: Beispiel OLE-DB-Ziel ohne schnelles Laden	65
Abbildung 25: Vermeidung der SSIS-Komponente Union All	67
Abbildung 26: Verwendung der statischen Analyse	75
Abbildung 27: Screenshot der Anwendung für die statische Analyse	76
Abbildung 28: Beispielhafte Darstellung der Auswertung in Excel	76
Abbildung 29: Anlegen eines Fehlers in der SharePoint-Liste	80
Abbildung 30: Übersicht über angelegte Fehler in der SharePoint-Liste	80
Abbildung 31: Sortierung der Fehlerliste	81

Tabellenverzeichnis

Soweit nicht anders vermerkt, sind die Tabellen eigene Darstellungen.

Tabelle 1: Einfaches Beispiel für Äquivalenzklassen.....	35
Tabelle 2: Erweiterung um Grenzwerttest	36
Tabelle 3: Kategorisierter Fragenkatalog mit Einschätzung des Testbedarfs... 47	
Tabelle 4: Testverfahren für testintensive und testarme BI-Projekte	48
Tabelle 5: Funktionsübersicht für den XML-Parser	62
Tabelle 6: Beispiel für eine Klassifizierung nach Auswirkungen von Datenqualitätsproblemen.....	78

Listings-Verzeichnis

Soweit nicht anders vermerkt, sind die Listings eigene Darstellungen.

Listing 1: Beispielhaftes SQL-Kommando für Fremdschlüssel-Tests	28
Listing 2: Beispiel für auszulagernde Parameter	57
Listing 3: MSBuild-Konfiguration.....	59
Listing 4: Auszug aus der MSBuild-Steuerdatei „BI.Quality.proj“	59
Listing 5: Batchdatei zum automatisierten Starten mit MSBuild.....	60
Listing 6: Inhalt des Script Task zur Ausführung des Test-Projektes.....	60
Listing 7: Auszug der XML-Datei für die Variante der Schnellauswahl	64
Listing 8: Auszug der XML-Datei für die Auswahl des SQL-Kommandos.....	64
Listing 9: Datenbankabfrage für alle Sichten	65
Listing 10: Auszug der XML-Datei für ausgewähltes schnelles Laden	66
Listing 11: Auszug aus der XML-Datei für das zu vermeidende Muster	67
Listing 12: Einführung in das DTSX-Format	68
Listing 13: Öffnen der XML-Datei	69
Listing 14: Suchen von Datenfluss-Komponenten	70
Listing 15: Auswertung von Datenfluss-Komponenten	70
Listing 16: LINQ-Abfrageausdruck zur Ermittlung einer Eigenschaft	71
Listing 17: SQL-Kommando-Parser mit ScriptDom	71
Listing 18: Beispiel für Verwendung eines regulären Ausdrucks	72
Listing 19: Beispielhafte Definition von Parser-Tests in XML.....	74

Abkürzungsverzeichnis

ADAPT	A pplication D esign for A nalytical P rocessing T echnologies
BI	B usiness I ntelligence
C#	C-Sharp (Programmiersprache von Microsoft)
CD	C ompact D isc
CSV	C omma S eparated V alues
DB	D aten b ank
DLL	D ynamic L ink L ibrary
DTSX	D ata T ransformation S ervices Package (Dateiformat)
DWH	D ata W arehouse
ERP	E nterprise R esource P lanning
ETL	E xtraktion, T ransformation, L aden
GB	G iga B yte
GUI	G raphical U ser I nterface (Grafische Benutzeroberfläche)
HTML	H ypertext M arkup L anguage
ISO	International Organization for Standardization
IT	I nformation s technologie
KMU	K leine und m ittlere U nternehmen
KPI	K ey P erformance I ndicator
LINQ	L anguage- I ntegrated Q uery (Sprachintegrierte Abfrage)
MDX	M ultidimensional E xpressions
MOLAP	M ultidimensionales OLAP
OLAP	O n- L ine A nalytical P rocessing
OLE DB	O bject L inking and E MBEDDING D ATABASE
PL/SQL	P rocedural L anguage/ SQL
ROLAP	R elationales OLAP
SCD	S lowly C hanging D imension
SQL	S tructured Q uery L anguage
SSIS	SQL S erver I ntegration S ervices
XML	E xtensible M arkup L anguage
ZIP	Dateiformat für komprimierte Dateien

1 Einleitung

Bestandteil eines jeden Software-Entwicklungs-Projektes ist das Testen. Doch wie wird das in Business Intelligence (BI) Projekten gehandhabt? Gerade für BI-Lösungen ist die Produktqualität ein kritischer Erfolgsfaktor, denn die Anwender müssen den Ergebnissen vertrauen können. Auch der wirtschaftliche Erfolg kann davon abhängen, wenn auf Grundlage der Ergebnisse wichtige Entscheidungen getroffen werden. Tests gewährleisten die Qualität der Entscheidungsunterstützung oder Informationsversorgung und decken Mängel der BI-Lösung auf.

„In der Praxis hat sich gezeigt, dass Mängel häufig durch eine fehlende oder ungenügende Ausschöpfung des Leistungspotenziales der BI-Anwendungssysteme verursacht werden. [...] Der Hauptgrund der Mängel liegt in der fehlenden oder unzureichenden Abstimmung der BI-Konzepte mit dem strategischen Management des Unternehmens.“¹

Zum Zeitpunkt der Erstellung dieses Buches existierte nur wenig Literatur zur Sicherstellung der Qualität von BI-Lösungen.

1.1 Aufgabenstellung und Zielsetzung

Projektleiter, Testmanager und Entwickler von BI-Lösungen werden mit diesem Buch mit neuen praxisrelevanten Testverfahren vertraut gemacht. Ihre Anwendung wird beschrieben und die Handhabung vereinfacht. Testverfahren aus der Software-Entwicklung werden auf BI-Projekte übertragen. Die Anforderungen an die Testverfahren im BI-Umfeld werden, genauso wie geeignete Testwerkzeuge und eine praktische Anwendbarkeit, berücksichtigt. Eine Verbesserung der Datenqualität steht nicht im Mittelpunkt.

1.2 Vorgehen

Zu Beginn des Buchprojektes wurden die Testverfahren aus der Software-Entwicklung studiert und international vorhandene geeignete Testkonzepte und Testverfahren aus dem BI-Umfeld gesucht. Im nächsten Schritt wurde versucht, die Testverfahren aus der Software-Entwicklung sinnvoll und praktisch verwertbar

¹ Kemper et al. 2010, S. 171 f.