

Die professionelle Access-Anwendung

Karl Donaubauer

Einleitung

Gründe und Ziele

Idee

- **AEK 14: "Status von Access"**
- **Image, Stellung, Modernität etc.**
- **Lothar Krahforst:**

Webseite parallel zur FAQ

**Was macht die professionelle
Access-Anwendung aus?**

Access-Image

- **beim Feind (IT-Abteilung, Admin, Developer...)**
 - jeder kann viel zu einfach unternehmenskritische Daten und Anwendungen...
 - chaotisch/anarchisch/viral
 - laienhaft erstellt
 - nix dokumentiert
 - halten sich nicht an Standards
- **zum Teil berechnete Kritik, Problem:**
 - Image überträgt sich auf das komplette Tool und **ALLE**, die damit arbeiten → Klischee

Typischer Access-Entwickler

- **Quereinsteiger**
- **Abteilungsleiter/Sachbearbeiter**
 - Anwendungen für eigene betriebliche Praxis
- **Einzelkämpfer, kleine Abteilung**
- **wenige professionelle Tools**
- **wenig Budget**
- **wenig Fortbildung im Fachbereich**

Microsoft®

- **Tool für Enduser, Poweruser**
- **Office-Entwickler <> Professionelle Entwickler**
 - prägt Image nach innen und außen
- **mangelnde Entwickler-Betreuung**
- **keine Entwickler-Tools von MS**
- **kein Entwickler-Marketing von MS**
- **manche Entwickler-Features entfernt**
 - Quellcode-Verwaltung, ADP, Replikation, Sicherheit

Gründe – Positiv

- **Access hat seinen Platz in jeder Organisation**
- **So viele professionelle Anwendungen wie nur wenige andere Tools**
 - große Branchensoftware
 - unzählige mittlere und kleinere großartig funktionierende Anwendungen
- **Viele professionell agierende Access-Entwickler**

Gründe – Positiv

- **Wir müssen eigentlich nix beweisen**
- **Sollten es aber doch**
 - um diese Dinge über die Rampe zu bringen
 - **Stundensätze und Gehälter erhöhen**
- **Defensive:**
 - "... auch mit Access kann man..."
 - nicht zielführend
- **Offensive: Katalog**

Ziele

- **Referenz**
 - eigene Einschätzung
 - Fremdeinschätzung
 - Kollegen, Gutachter, (potentielle) Kunden
- **Richtlinien**
 - eigene Arbeit
 - Deklaration wichtiger als "Tipp"-Charakter
 - Selbstverständlichkeiten statt Sensationen
- **Image-Verbesserung**
 - Vergleichbares für andere Entwickler-Plattformen?

Gründe+Ziele persönlich

- 1994-98 Access/Entwickler-Lehrjahre
- 1998-2003 Gesellenjahre
- Ab 40 keine Ausreden mehr!
- bzw. ab ~70 € Stundensatz

Gründe+Ziele persönlich

- **Überarbeiter → Berater → Gutachter**
 - **Katalog könnte helfen**
 - Schema für schnellen, geordneten Überblick
 - "genormte" Beurteilungskriterien für Analyse
- **Beschäftigung mit Teilaspekten**
 - **Design-Vortrag AEK3**
 - **Qualitätsmanagement**
 - **Projektmanagement**

Legitimität

- **AEK = 200 Access-Entwickler**
 - Vortrag + Diskussion
 - Fragebogen
 - www.donkarl.com/forum
 - office@donkarl.com

- Organisation der
Datenbankentwickler



- **11 aktive/ehemalige MS MVPs für Access**
Höchst-Wertvolle-Professionelle

Überblick

Software-Qualitätsmanagement

Software-QM

- QM-Abteilungen, Qualitätsmanager
- Terminologie
 - Qualitätsorganisation, Qualitätspolitik
 - Qualitätsmanagementsystem
 - Strukturen, Prozesse etc. etc.
- Klischee:
SW-QM ist etwas für große Unternehmen

STIMMT!

QM und Access

- **Prinzipien, Definitionen, Normen**
 - Kosten/Nutzen, "Qualität", ISO...
- **Methoden, Techniken**
 - Komplexität messen
 - Testen
 - Agile Methoden

Qualitätsmanagement

- **Prozessmanagement**
 - Requirements Engineering
 - Testmanagement
 - Konfigurationsmanagement
 - Usability Engineering
- **Wissensmanagement**
 - Nutzung, Speicherung, Kommunikation von Erfahrungen (z.B. Wissensdatenbanken)
- **QM auf Projektebene**
 - praktischer Qualitätsplan

QM – Kosten und Nutzen

- **Maßnahmen müssen sich lohnen**

Kosten für QM-Maßnahmen

gg.

Kosten durch mangelnde Qualität

Fehlervermeidungskosten

gg.

Fehlerkosten

- **Weiche Faktoren**
 - Vermitteln von Zuverlässigkeit
 - Qualitäts-Image

Qualitätsmanagement

- **Produktqualität**

- Hat das Produkt best. Eigenschaften?
- Erfüllt es seinen Zweck und wie gut?
- Katalog zielt v.a. auf Produktqualität

- **Prozessqualität**

- gute Prozesse → gutes Produkt
- ...Anforderungen...Spezifikationen...Entwurf...
Konfiguration...Entwicklung...Test...Installation...
Schulung...Wartung...

"Qualität"

Qualität ist erstens die Fähigkeit vorherzusagen, was morgen, nächsten Monat und nächstes Jahr geschehen wird.

Und es ist zweitens die Fähigkeit, später zu erklären, warum alles ganz anders gekommen ist.

(Winston Churchill)

Qualität

- **Die Summe aller relevanten Eigenschaften eines Software-Produkts, mit denen seine Kunden zufriedengestellt werden, und**
- **die Summe der dazu notwendigen Eigenschaften von Software-Prozessen wie z.B. erreichte Reifegrade, die zur Erstellung, zum Betrieb und zur Pflege gefordert werden.**

Ernest Wallmüller

Software Quality Engineering, 2011, S 10

Qualitätsmodell

- **Qualitätsanforderungen beim Kunden abfragen**
- **Priorisieren und Konkretisieren**
- **Anforderungen auf 3 Ebenen anordnen**
 - Ziele
 - Aspekte, Kriterien
 - Indikatoren, Metriken
- **Qualitätsmodell ist Referenz für**
 - Entwicklung
 - Reviews
 - Abnahme beim Kunden

Qualität fassbar machen

- **Abstrakte Qualitätsziele**
 - Kunde: Hochverfügbarkeit
- **Konkrete Qualitätsaspekte**
 - Robustheit gg. Überlastung, gg. Fehlbedienung
- **Indikatoren und Metriken**
 - 100 DS/Sekunde
 - 20 konkurrierende Anwender

Code-Metriken

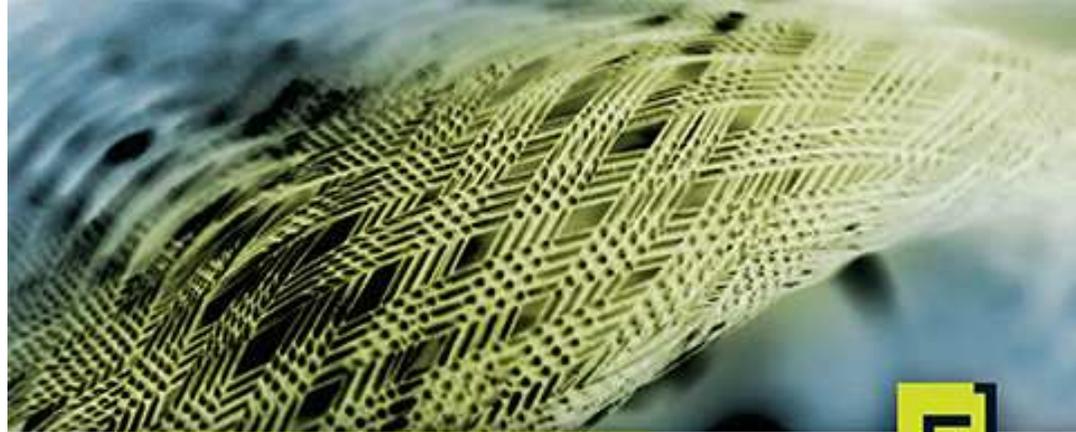
- **Lines of Code**

- nur Zeilen mit Befehle
 - mehrzeilige Befehle
 - mehrere Befehle:Zeile
- Kommentare
- Leerzeilen

```
1 Private Sub Form_Current()  
2  
3   On Error GoTo myError  
4  
5   Dim rs As DAO.Recordset  
6   Set rs = CurDbC.OpenRecordset("SELECT ArbPN:  
7  
8   'pruefen, ob ArbPName mit LG existiert --> nicht ko  
9   If Me!ArbPName Like "*"LG" Then  
10      Me!IbINichtKommissioniert.Visible = False  
11      Exit Sub  
12   End If  
13  
14   'nicht kommissionierte Ware kennzeichnen  
15   Me!IbINichtKommissioniert.Visible = Not rs.EOF  
16  
17 myExit:  
18   'aufraeumen  
19   If Not rs Is Nothing Then rs.Close: Set rs = Nothing  
20   Exit Sub
```

- **Zyklomatische Komplexität
(McCabe-Metrik, CC)**

- Punkt für jede Verzweigung, Schleife
- $CC < 10 \rightarrow$ gut testbar \leftrightarrow verständlich



→ 2., überarbeitete und erweiterte Auflage



Kurt Schneider

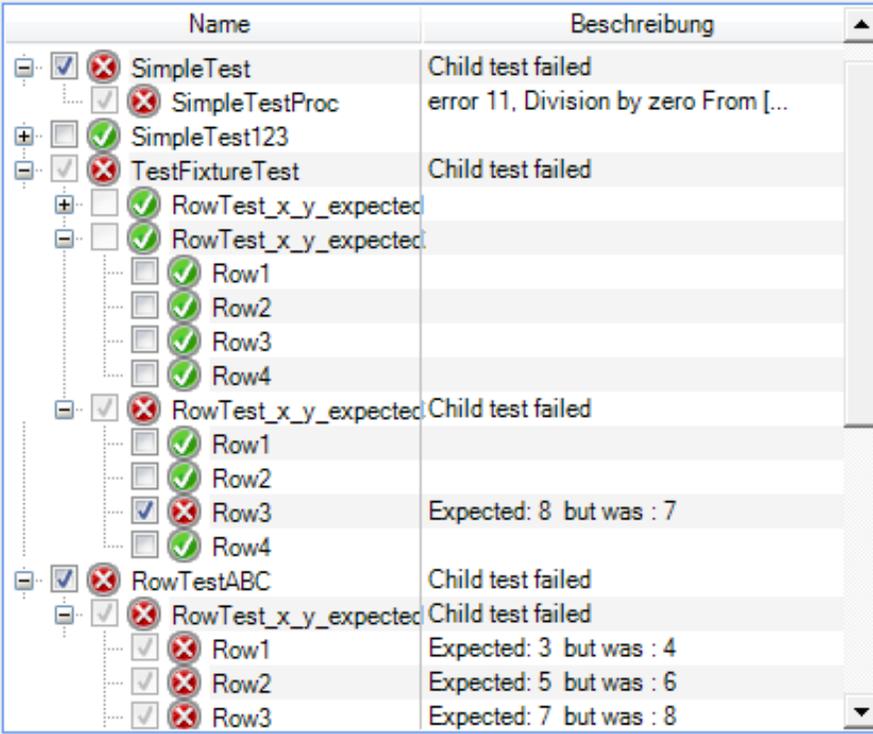
Abenteuer Software Qualität

Grundlagen und Verfahren für Qualitätssicherung
und Qualitätsmanagement

dpunkt.verlag

Testen

- im Access-Bereich vernachlässigt
 - nichts integriert vom Hersteller MS
 - lange Zeit keine Werkzeuge
 - Josef Pötzl: AccUnit
<http://accunit.access-codelib.net>
- AEK13:
Pragmatisches Testen,
Paul Rohorzka



Name	Beschreibung
SimpleTest	Child test failed
SimpleTestProc	error 11, Division by zero From [...]
SimpleTest123	
TestFixtureTest	Child test failed
RowTest_x_y_expected	
RowTest_x_y_expected	
Row1	
Row2	
Row3	
Row4	
RowTest_x_y_expected	Child test failed
Row1	
Row2	
Row3	Expected: 8 but was : 7
Row4	
RowTestABC	Child test failed
RowTest_x_y_expected	Child test failed
Row1	Expected: 3 but was : 4
Row2	Expected: 5 but was : 6
Row3	Expected: 7 but was : 8

Reviews

- **Prüfverfahren: ein Dokument, mehrere Gutachter**
- **Konzentration auf ausgewählte Aspekte**
- **Ergebnisse in gemeinsamen Sitzungen zusammengeführt**
- **vom lockeren Walkthrough bis zur streng regulierten technischen Inspektion**
- **setzt früh an, i.d.R. vor dem Testen → spart viel Geld**
- **Fehlerkosten steigen exponentiell über die Verweildauer des Fehlers im Projekt**

Agile Methoden

- **2001 Agiles Manifest**
- **Bewegung gegen**
 - umfangreiche, "reife"/starre Prozesse
 - aufwändige Dokumentation
 - monatelange Vorausplanung
 - Bürokratie
- **überschaubar, unmittelbar, flexibel, kommunikativ**
- **hoher Qualitätsanspruch**
- **vieles mit Access anwendbar**
- **kein Allheilmittel, abhängig vom Projekttyp, Umfeld...**
- **Extreme Programming, Scrum, Lean, Kanban**

Extreme Programming (XP)

- **Codequalität steigern**
 - ständige Reviews – Pair Programming
- **Kundenzufriedenheit steigern**
 - früh und oft lauffähigen Code liefern
→ Feedback, Kommunikation
 - Kunde eng eingebunden (On-Site Customer)
- **Anforderungen → Story Cards**
- **Aufwandsschätzung → Planning Game**
- **Unit Tests, Testgetriebene Entwicklung, Refactoring...**

Pair Programming

- **2 Entwickler an 1 Rechner**
- **1 tippt, 1 kontrolliert und denkt weiter**
- **ständige gegenseitige Korrekturen und Besprechungen**
- **alle paar Minuten Wechsel an der Tastatur**
- **Zusammensetzung des 2er-Teams wechselt**
 - **Code-Konventionen**
 - **häufiges Integrieren der Ergebnisse**

XP und SW-Qualität

- **Qualität vor Funktionsumfang**
 - Zeit knapp oder Ressourcen knapp
 - Qualität halten, Funktionalität weglassen
- **Kundenzufriedenheit ist das wichtigste Qualitätskriterium**
 - andere Prinzipien aus dem klassischen SW-QM werden ihr untergeordnet
- **Testen hat extrem hohen Stellenwert**

Agiles Projektmanagement

- **Scrum**
 - fixe, kurze Entwicklungszyklen (Sprints)
 - AEK13: Scrum, Carola Hojnacki
- **Kanban**
 - Arbeitsabläufe visualisieren
 - Zahl der Aufgaben im System begrenzen
 - Aufgaben holen statt reindrücken (Pull statt Push)
 - Engpässe beseitigen, kürzere Durchlaufzeiten, bessere Vorhersagbarkeit
 - AEK16 ?

Prozessqualität auf AEKs

1998 Marketing

1999 Pflichtenheft

2000 Design

2001 EDV-Verträge

2008 Konfigurationsmanagement

2009 Produktionsprozess

2010 Scrum, Testen

2011 Geschäftsregeln

2012 Spec-by-Example, wir hier...

Hauptteil

Access-Katalog

Bereiche

- **Datenbank**
 - JET/ACE, ODBC
- **Anwendung(sorganisation)**
 - Dateien, Access-Objekte, GUI
- **Programmierung**
 - VBA, Makros

Strenge

- **Muss**

- Ausnahmen nur mit sehr guter Begründung
- Ausreden nicht erlaubt

- **Soll**

- Empfehlung
- kann evtl. auch anders gelöst werden
- Ausnahmen mit Begründung
- gute Ausreden erlaubt

Methode

- **Beschreibung**
 - **Stichworte**
 - **Beispiele**
-
- **Begründung**
 - **Referenzen, Links**

Begründungen

- **Technische Vorteile**
 - Datenkonsistenz
 - Performance
 - Robustheit
- **Wartbarkeit**
 - Klarheit, Lesbarkeit, Effizienz
 - Verständlichkeit für andere Entwickler
- **Fehlerprävention**

Bereich

Datenbank

Normalisierung

- **0. bis 3. Normalform wird mind. eingehalten**
 - 0. = keine Berechnungen speichern
 - 1. = atomarer Wert pro Feld
 - 2. = Nichtschlüselfelder abhängig vom kompletten Schlüssel
 - 3. = Nichtschlüselfelder direkt abhängig vom Schlüssel
- **Datenkonsistenz**
- **AEK 7/8, Normalisierung, Michael Zimmermann**

Primärschlüssel

- Jede Tabelle hat einen Primärschlüssel

- Identifizierung bei allen Gelegenheiten
- Aktualisierbarkeit
- JOINS

Indizierung

- **Jede Tabelle ist adäquat indiziert**
 - Indizes für alle Felder und Feldkombinationen, über die verknüpft, gefiltert, sortiert wird
 - die Indizes sind so restriktiv wie möglich wenn eindeutig möglich, dann Ohne Duplikate
- **Performance**
- **Performance**
- **Performance**
- **erlaubte Ausnahme: Performance**
 - massenhaft INSERT, UPDATE, DELETE

Beziehungen und RI

- **Beziehungen für Tabellen sind gesetzt**
- **Referentielle Integrität ist aktiviert**
- **Aktualisierungs- und Löschweitergabe ist aktiviert, wenn inhaltlich möglich/sinnvoll**

- **Datenkonsistenz**
- **Überblick, Tabellendesign-Kontrolle**

Benennung 1

- **Keine SQL/VBA/Access – Schlüsselworte in benutzerdefinierten Benennungen**
 - Tabellen, Abfragen, Felder, Spalten, Aliase
- eckige Klammern nötig
- Fehlverhalten trotz [] möglich
- Schlüsselwortliste von Allen Browne
<http://www.allenbrowne.com/ApplyBadWord.html>

Benennung 2

- **Keine Leer- oder Sonderzeichen (inkl. Umlaute) in benutzerdefinierten Benennungen**
 - Tabellen, Abfragen, Felder, Spalten, Aliase
- eckige Klammern nötig
- Probleme in fremdsprachigen Umgebungen
- Probleme mit Tools (Upgrade, Konvertierung)

Benennung 3

- **Durchgängige Benennungsregeln**
 - Tabellen, Felder, Abfragen
 - Namenskonvention oder eigenes System
 - Objekt- und Datentypen und/oder Funktion (Fremdschlüssel...) erkennbar

- **Wartbarkeit**
- **Fehlerprävention**

Abfragen sind optimiert

- **Indexnutzung wo immer möglich**
 - Vermeidung von Table Scans
 - Nutzung der optimalen Indexvariante
 - JOINS/Filter/Sortierung über indizierte Felder
- **nur benötigte Datensätze ausgeben (WHERE)**
- **nur benötigte Spalten im SELECT**
- **nur notwendige Sortierungen**
- **D-Funktionen durch Tabelleneinbindung vermeiden**

- **AEK8, Performance in Abfragen, Michael Zimmermann**

ODBC-Abfragen sind optimiert

- **Grundlegende Performanceregeln werden eingehalten**
 - lokale Verarbeitung durch JET/ACE vermeiden
 - keine Access-spezifischen Funktionen
 - keine Bezüge auf Access-Objekte (Formularfelder)

- **Performance**

Arbeit mit ODBC-BE optimiert

- **Performancekritische Abfragen sind auf den Server verlagert**
 - Pass Through-Abfragen
 - Server-Objekte nutzen
 - Stored Procedures, Views...
- **Performance**
- **AEK11, Migration auf SQL Server, Bernd Jungbluth**

Datenbank

1. Normalisierung
 2. Primärschlüssel
 3. Indizierung
 4. Beziehungen + RI
 5. Keine Schlüsselworte
 6. Keine Sonderzeichen
 7. Namenskonvention
 8. Abfragen optimiert
 9. ODBC-Abfragen optimiert
 10. Arbeit mit ODBC-BE optimiert
- Was fehlt im Bereich DATENBANK?
 - Was muss raus oder ist schlecht formuliert?
 - Passen **Muss/Soll**?

Bereich

Anwendung

Aufteilung in Frontend und Backend

- Tabellen liegen im BE (evtl. in mehreren),
alles andere im FE
- Jeder Nutzer erhält ein eigenes, lokales FE
(bei TS in einem Serverordner)

- Vermeidung von Datei- und Datenkorruption
- Effiziente Wartung des FEs
- Weniger Netzverkehr
- <http://www.donkarl.com?FAQ1.35>

Aktualisieren der Frontends

- In Mehrbenutzerumgebungen wird ein Mechanismus zum Verteilen neuer FEs verwendet.
 - nicht manuell auf jeden Arbeitsplatz verteilt
 - Tool, Skript, Batch etc.
- Effizienz
- Benutzer-/Admin-Freundlichkeit
- <http://www.donkarl.com?FAQ1.35>

Benennung 1

- **Keine Access/VBA/SQL – Schlüsselworte in benutzerdefinierten Benennungen**
 - Access-Objekte, Steuerelemente
- eckige Klammern nötig
- Fehlerprävention, Fehlverhalten trotz [] möglich
- Schlüsselwortliste von Allen Browne
<http://www.allenbrowne.com/ApplyBadWord.html>

Benennung 2

- **Keine Leer- oder Sonderzeichen (inkl. Umlaute) in benutzerdefinierten Benennungen**
 - Access-Objekte, Steuerelemente

- **eckige Klammern nötig**
- **Probleme in fremdsprachigen Umgebungen**

Benennung 3

- **Durchgängige Benennungsregeln**
 - Access-Objekte, Steuerelemente
 - Namenskonvention oder eigenes System
 - Objekt/Steuerelement-Typen erkennbar

- **Wartbarkeit**
- **Fehlerprävention**

Komprimierung

- **JET/ACE-Backends und Frontends werden regelmäßig komprimiert**
 - Objekt- und Datenmüll gelöscht
 - Tabellen defragmentiert
 - Abfragestatistiken erneuert

- **Microsoft Jet Database Engine Programmer's Guide - Chapter 4**

Performance-Maßnahmen

- **Wichtige und bekannte Performance-Maßnahmen sind berücksichtigt**
 - Objektnamensautokorrektur deaktiviert
 - Unterdatenblätter deaktiviert
 - Permanentes Recordset bei JET/ACE-BE
 - Keine D-Funktionen auf eingebundene Tabellen
- Performance-Webseiten von FMS und Tony Toews
 - <http://www.fmsinc.com/tpapers/faster/>
 - <http://www.granite.ab.ca/access/performancefaq.htm>

Tote Objekte

- **Nicht mehr in Verwendung stehende Objekte sind entfernt**

- **Wartbarkeit**
- **Speicherplatz, -optimierung**

Ergonomie

- **ISO 9241 zur Software-Ergonomie ist grundlegend erfüllt**
 - Aufgabenangemessenheit
 - Selbstbeschreibungsfähigkeit
 - Steuerbarkeit
 - Erwartungskonformität
 - Fehlertoleranz
 - Individualisierbarkeit
 - Lernförderlichkeit
- **AEK3: Design, Karl Donaubauer**

Benutzeroberfläche

- **Der Anwender arbeitet nur in der für ihn gestalteten Benutzeroberfläche**
 - sieht nur Formulare, Berichte, Menüs, Ribbons etc.
 - sieht keine Tabellen, Abfragen, Codes, Entwurfsansichten

- **Benutzerfreundlichkeit**
- **Stabilität**

Anwendung

1. Aufteilung in FE und BE
2. FE-Aktualisierung automatisiert
3. Indizierung
4. Beziehungen und RI
5. Keine Schlüsselworte
6. Keine Sonderzeichen
7. Namenskonvention
8. Komprimierung
9. Wichtige Performance-Maßnahmen
10. Tote Objekte entfernt

- Was fehlt im Bereich ANWENDUNG?
- Was muss raus oder ist schlecht formuliert?
- Passen **Muss/Soll**?

Bereich

Programmierung

VBA vor Makros

- VBA hat Vorrang
- Makros nur in Spezialfällen
 - Autoexec
 - Autokeys
 - Datenmakros
 - Webdatenbanken

- Leistungsumfang, Flexibilität, Effizienz von VBA größer

Benennung 1

- **Keine VBA/Access/SQL – Schlüsselworte in benutzerdefinierten Benennungen**
 - Prozeduren, Variablen, Konstanten
- Fehlerprävention
- **Schlüsselwortliste von Allen Browne**
<http://www.allenbrowne.com/ApplyBadWord.html>

Benennung 2

- **Keine Umlaute in benutzerdefinierten Benennungen**
 - **Prozeduren, Variablen, Konstanten**

- **Probleme in fremdsprachigen Umgebungen**

Benennung 3

- **Durchgängige Benennungsregeln**
 - Prozeduren, Variablen, Konstanten
 - Namenskonvention oder eigenes System
 - Datentypen sind erkennbar

- **Wartbarkeit**
- **Fehlerprävention**
- **Reddick Namenskonventionen**
<http://www.xoc.net/standards/>

Variablendeklaration

- **Alle Variablen sind mit Datentypen deklariert**
 - Option "Variablendeklaration erforderlich"
 - Option Explicit

- **Wartbarkeit, Fehlerprävention**

Verweise

- Die Verweise sind auf die tatsächlich im Programm verwendeten reduziert
- Fehlerprävention
- Performance
- <http://www.donkarl.com?FAQ7.1>

SQL vor DAO/ADO

- Für Datenoperationen werden Abfragen/SQL verwendet, außer Programmierung per DAO/ADO ist nötig
 - nötig z.B. bei Schleifen mit Zwischenspeicherung, Berechnung
- Performance bei Massenoperationen
- Sperrung durch Recordset
- Fehlerprävention – Code potentiell fehlerträchtig

Fehlerbehandlung

- **Ausreichende Fehlerbehandlung ist vorhanden**
 - in der Runtime in jeder Prozedur

- **Stabilität**
- **Benutzerfreundlichkeit**
- **<http://www.mztools.com/v3/mztools3.aspx>**

Kommentare

- **Code ist ausreichend kommentiert**
 - zu Beginn nicht trivialer Prozeduren ist ihr Zweck erläutert
 - alle nicht trivialen Codestellen soweit, dass sie ein sachkundiger Entwickler versteht

- **Wartbarkeit**

Formatierung

- **Code ist lesefreundlich formatiert**
 - **Einrückungen**
 - horizontale Gliederung bei Schleifen, Kontrollstrukturen etc.
 - **Leerzeilen**
 - vertikale Gliederung nach inhaltlichen Kriterien
 - **Zeilenumbrüche**
 - bei überlangen Zeilen

- **Wartbarkeit**

Programmierung

1. VBA vor Makros
 2. Keine Schlüsselworte
 3. Keine Umlaute
 4. Namenskonvention
 5. Variablendeklaration
 6. Verweise
 7. SQL vor DAO/ADO
 8. Fehlerbehandlung
 9. Kommentare
 10. Formatierung
- Was fehlt im Bereich PROGRAMMIERUNG?
 - Was muss raus oder ist schlecht formuliert?
 - Passen **Muss/Soll**?

Diskussion

- **Prozesse und Drumherum in den Katalog?**
 - seriöse Anforderungserhebung
 - Tests
 - Doku, Hilfe
 - Konfigurationsmanagement, Setup
- **Bewusst keine Stil- und Glaubensfragen**
 - Klassen oder nicht
 - D-Funktionen oder nicht

zu tun für mich

- **Katalog V1.0 fertig stellen**
 - Punkte, Begründung etc. ausformulieren
 - Referenzen erweitern
- **Veröffentlichen**
 - www.donkarl.com
- **Propagieren**
 - AEK 16

zu tun für euch

- **Am Katalog mitarbeiten**
 - www.donkarl.com/forum
 - office@donkarl.com
 - Propagieren
- **Prüfen der eigenen Anwendungen**
- **Überlegen, Durchführen von Verbesserungen**
 - Produktqualität, Prozessqualität
 - Erfahrungsaustausch mit Entwickler-Kollegen
 - AEK, Stammtische
 - Review, Pair Programming...

Referenzen und Ressourcen

AEK14: **Status von Access**, Karl Donaubaueer/Paul Rohorzka
http://www.donkarl.com/Downloads/AEK/AEK14_Status.zip

AEK13: **Pragmatisches Testen**, Paul Rohorzka
http://www.donkarl.com/Downloads/AEK/AEK13_Testen.zip

Josef Pötzl: **AccUnit**
<http://accunit.access-codelib.net>

AEK13: **Scrum**, Carola Hojnacki
http://www.donkarl.com/Downloads/AEK/AEK13_Scrum.zip

AEK11: **Migration auf SQL Server**, Bernd Jungbluth
http://www.donkarl.com/Downloads/AEK/AEK11_Migration.zip

AEK8: **Performance in Abfragen**, Michael Zimmermann
http://www.donkarl.com/Downloads/AEK/AEK8_Abfragen_Performance.zip

AEK7: **Normalisierung**, Michael Zimmermann
http://www.donkarl.com/Downloads/AEK/AEK7_Normalisierung.zip

AEK3: **Design**, Karl Donaubaueer
http://www.donkarl.com/Downloads/AEK/AEK3_Design.zip

Referenzen und Ressourcen

FMS Performance-Tipps

<http://www.fmsinc.com/tpapers/faster/>

Tony Toews Performance-Tipps

<http://www.granite.ab.ca/access/performancefaq.htm>

Allen Browne Schlüsselwortliste

<http://www.allenbrowne.com/AppIssueBadWord.html>

Reddick Namenskonventionen

<http://www.xoc.net/standards/>

MZTools für VBA

<http://www.mztools.com/v3/mztools3.aspx>

Abenteurer Softwarequalität

Grundlagen und Verfahren für Qualitätssicherung und Qualitätsmanagement

Kurt Schneider, dpunkt-Verlag, Heidelberg 2012, ISBN 978-3-89864-784-7

Software Quality Engineering

Ein Leitfaden für bessere Software-Qualität

Ernest Wallmüller, Hanser, München 2011, ISBN 978-3-446-40405-2