

# Entwurfsmuster mit Access

Best Practices aus einer stark frequentierten Access-Praxis

Karl Donaubauber

AEK19

# Worum geht's?

- Vorstellung
  - Was mache ich mit Access?
- Richtlinien
  - Für den Access-Fundi
- Entwurfsmuster
  - Sichtbare Anwendungen
  - Unsichtbare Programme
  - Demos, Demos, Demos

Vorstellung

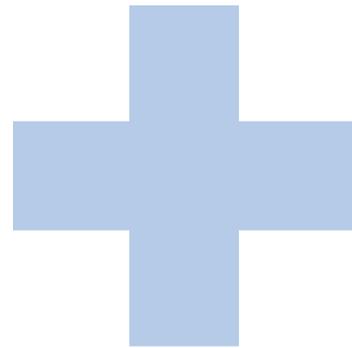
Was mocht i mit Access?

# Access-Praxis

- 20 Jahre
- 100 Access-Projekte
  - 50% eigene, 50% fremde
  - Offener Code, keine einzige MDE/ACCDE
  - Access = „offene Plattform“
  - Kunde = „Entwickler“
- Projekttypen
  - Neuerstellung
  - Optimierungen, Ergänzungen
  - Training, Expertisen

# Best Practices

Richtlinien



Entwurfsmuster

Richtlinien

für den Access-Fundi

# Defensiv entwickeln

- Keine externen Elemente
  - ActiveX, Add-Ins, Nicht-Standard-Verweise
- Keine „Halb-Externen“ Elemente
  - Ribbons, NaviBereich, Menüs, Symbolleisten etc.
  - Meine GUI (Formulare, Berichte) enthält ALLES
- Vorrang für in Access/ACE eingebaute Funktionalität
  - Nur programmieren, was nicht da ist

# Defensiv entwickeln

- Lange Wartezeit/Testphase bei Neuheiten
  - Access-Versionen, Service Packs, Bugfixes, Funktionalitäten
- Mit niedrigster Anwender-Version entwickeln
  - Kompatibilität, Lauffähigkeit
  - Verzicht auf neue Funktionen zugunsten von Stabilität
- Late Binding
  - Office-Versionen...
- Weitere Defensiv-Methoden
  - [Katalog zur professionellen Access-Anwendung](#)

# Ausnahmen von der Defensive

- Fehlende Funktionalität für konkrete Aufgabe
- Schwächen, Bugs von Access
- Häufig – Manchmal
  - Win-API: Datei-Dialog, Auflösungen, Positionen
  - Access-Boardmittel-Tools: [Application Starter](#), [Find&Replace](#)
- Gelegentlich – Selten
  - Access-Fremde-Tools/Bibliotheken: WinSCP, HTML/XML-OLBs
- Extrem Selten
  - ActiveX: Komplexe Barcodes

# Defensiv – Was bringt's?

- Stabil, verlässlich, Bug-arm
  - Große Anwendungen (oft) mit JET/ACE-Backend
  - Bis zu 100 Benutzer (TS)
- Versionswechsel, Upgrade
  - Access, Windows
  - MDB → ACCDB
  - 1-5 Minuten

# Einfach = KISS

- Architektur

- Transparente Struktur, Dateien, Beziehungen

- GUI

- Struktur, Navigation, Bedienung, Ergonomie

- Programmierung

- So wenig wie nötig
- Schlicht, keine Blackboxen

- Wartbarkeits-Grundsatz

- Halbwegs versierter Fremdentwickler soll Anwendung warten können

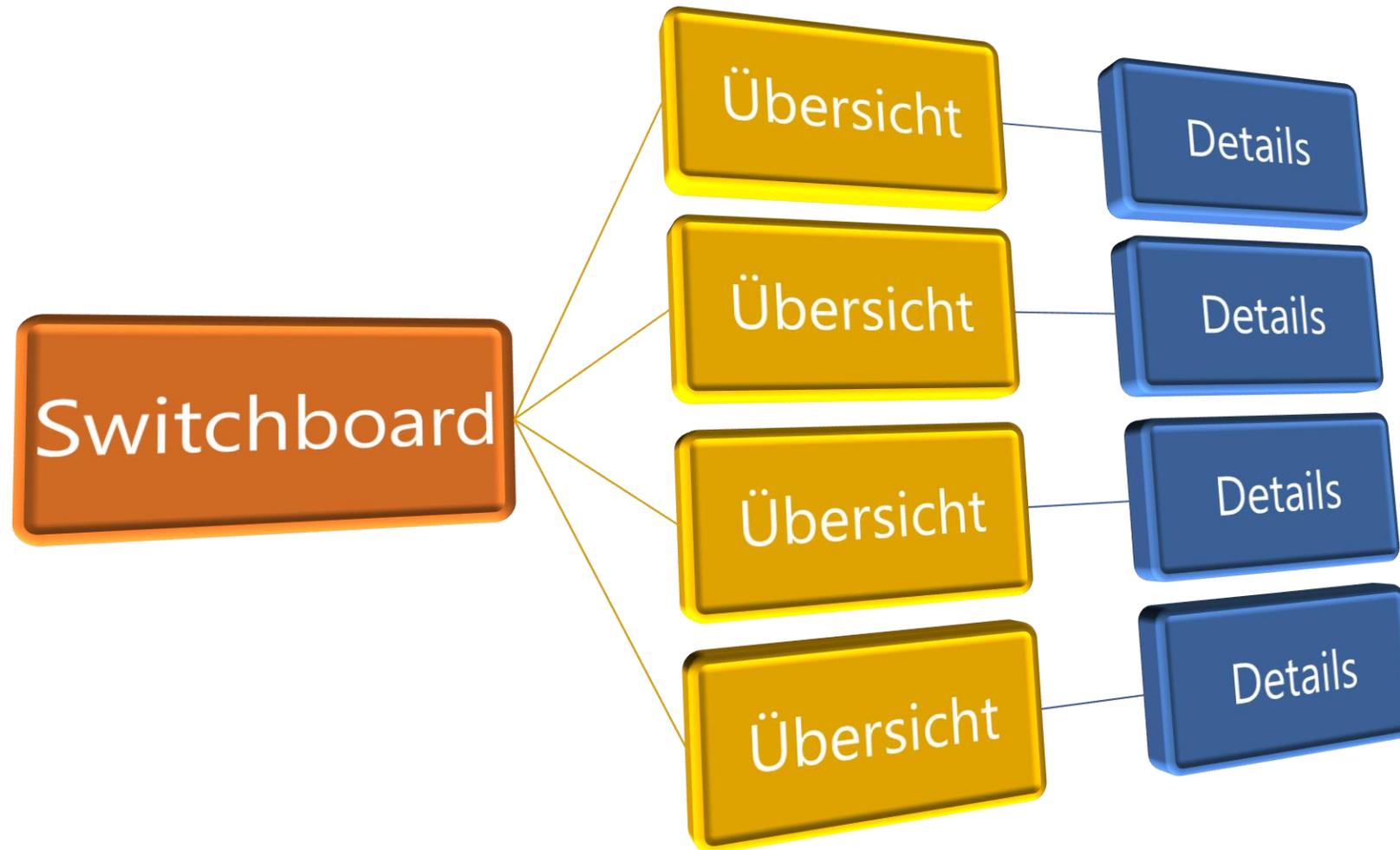
Entwurfsmuster

Sichtbare Anwendungen

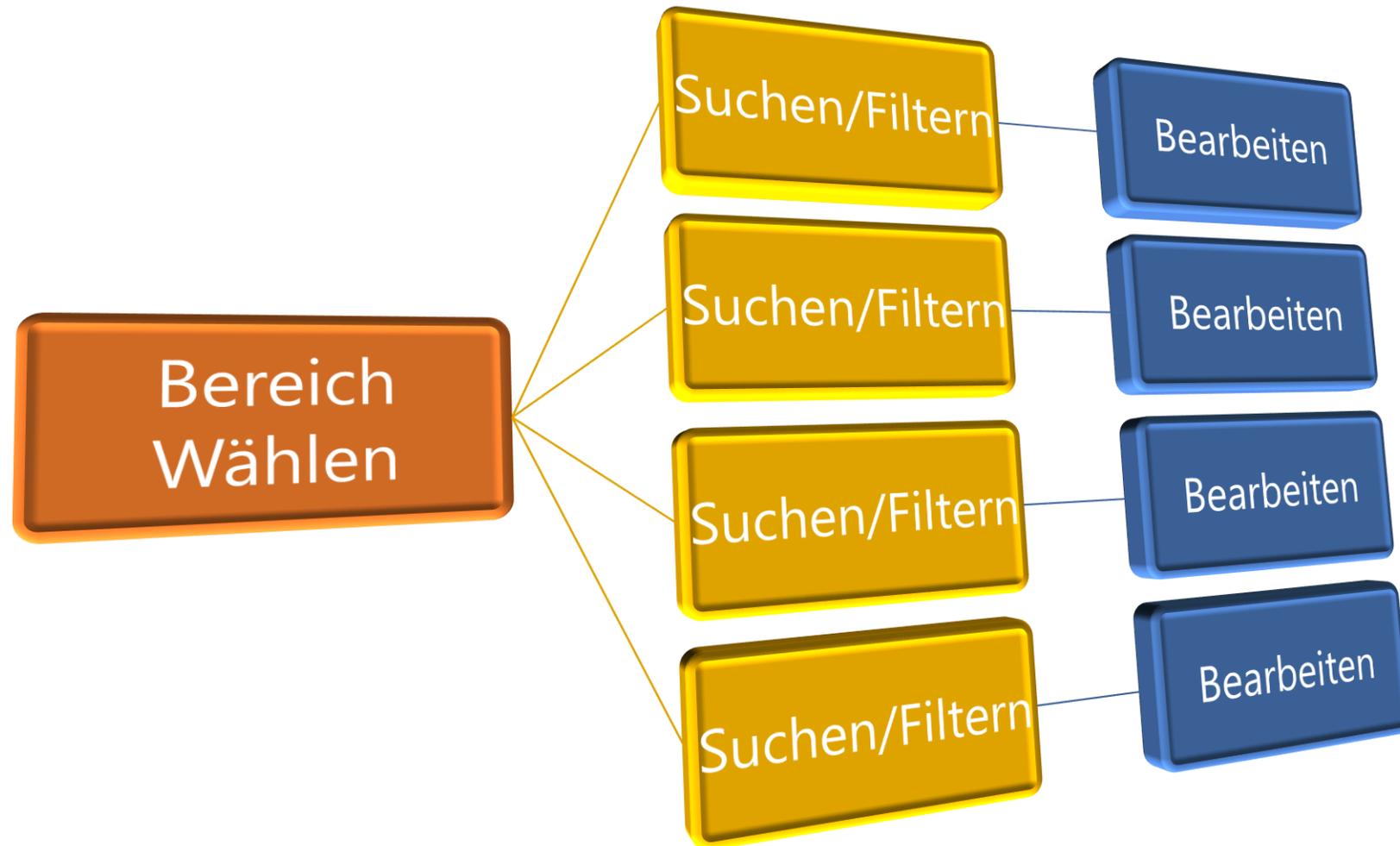
# Entwurfsmuster

- Struktur der Oberfläche (GUI)
  - Formularvorlagen/Schablonen
  - Abfragen, VBA-Code, Benennungen etc.
- Wiederverwendbare Teilbereiche/Tools
  - Zeitplan, Benutzersteuerung, Mehrsprachigkeit...
- 50-90% der Anwendung
- Anpassung
  - Datenquellen
  - Detailobjekte (UFos, Schaltflächen etc.)
  - Formatierungen (Größe, Position, Schrift etc.)

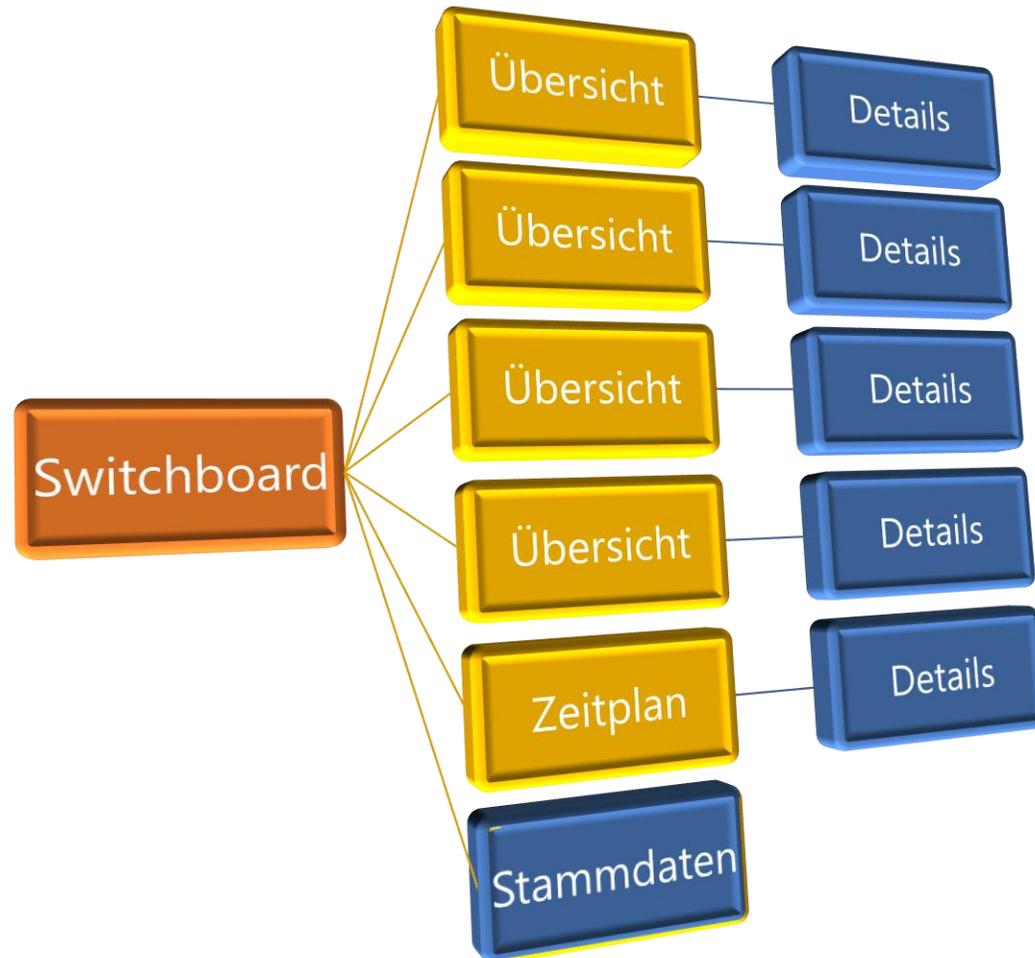
# Struktur der Oberfläche



# Struktur der Oberfläche



# Struktur der Oberfläche



# Entwurfsmuster – Vorteile

- Schnell, effizient im Umsetzen
- Robust, Bug-arm
- Ergonomie – gute Bedienbarkeit
- Sicherheit, Gewissheit, Herzeigeobjekte
- Kalkulierbar → Aufwand, Zeit, Kosten

# Kalkulierbar

- Standardbereich (Kunden, Lieferanten, Artikel...)
  - 2-4 Stunden
- Bereich + „Sonderwünsche“ (Kategorien, Berichte...)
  - 4-8 Stunden
- Einbau Teilbereiche
  - Zeitplan: 2-4 Stunden
  - Benutzersteuerung: 2-4 Stunden
  - Mehrsprachigkeit: 2-4 Stunden + abh. v. Größe/Anzahl

# Anforderungsprofil

- Systemvoraussetzungen
  - Betriebssystem(e), Netzwerk, Verbindungen
  - Office/Access-Version(en), Runtime
  - Anzahl Benutzer ...
- Leistungsumfang allgemein
  - „Organisiert werden soll ... “
  - ggf. Zeitrahmen, Budget
- Leistungsumfang detailliert: Bereiche + Datenbankobjekte
  - Kunden = Formular X, Bericht Y
  - Lieferanten = Formular X, Bericht Y
  - Anzahl + indiv. Aufwand → Kalkulation

Entwurfsmuster

Unsichtbare Programme

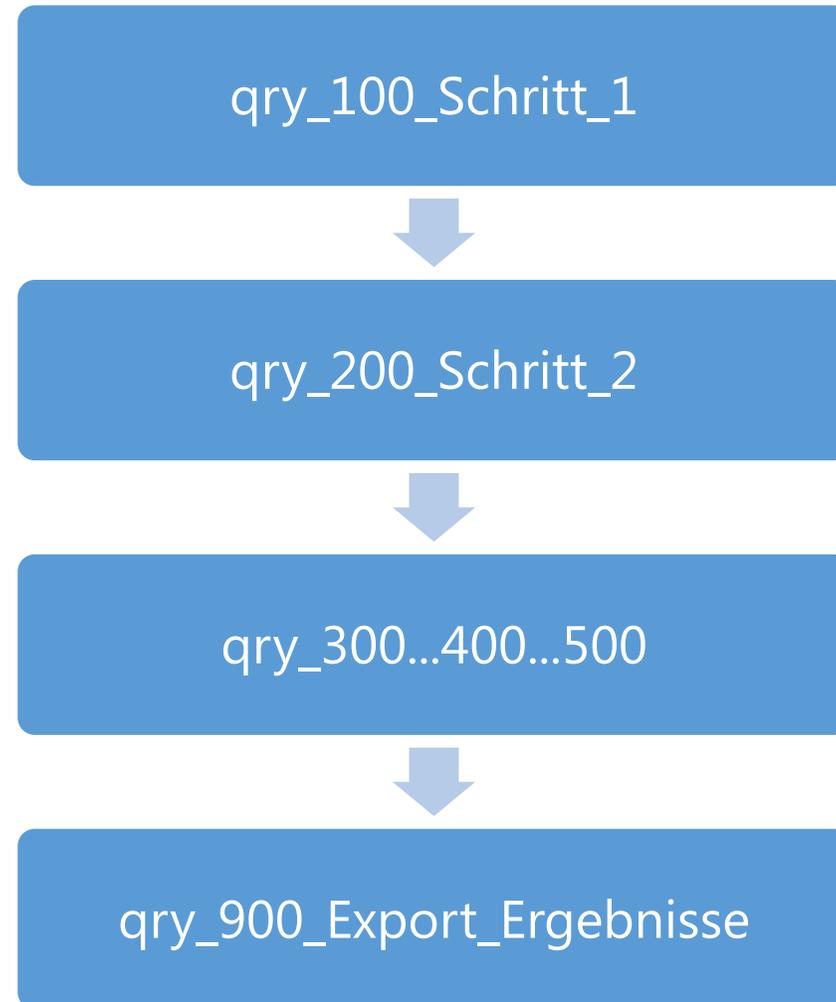
# Unsichtbare Programme

- Modellierung von Abläufen (Workflows)
  - Datenübernahmen
  - Verarbeitungen
  - Ergebnisse liefern/exportieren
  - Einmalig (z.B. initialer Datenimport)
  - Wiederkehrend → Windows-Tasks
- Keine Benutzer
  - Nur bez. Resultat (Ergebnisdaten, Auswertungen, Berichte)

# Entwurfsmuster

- Systematische/strukturierte Abarbeitung der Abläufe
- Durchnummerierte Aktionsabfragen
  - + zuarbeitende Auswahlabfragen
- Standardcode führt Abfragen aus
  - + individueller „Zwischen-“ Code für komplexe Verarbeitungen
  - z.B. Stücklistenauflösung, Parser für Excel Datenwürste

# Entwurfsmuster



# Entwurfsmuster – Vorteile

- Struktur, Code zur Modellierung vorhanden
- Anpassungsfähig
  - Nummerierung änderbar/erweiterbar
  - Umsortieren, einfügen, kopieren, SQL-Text umschreiben
- Übersichtlich
  - Gespeicherte Abfragen statt Code
  - Nummerierung+Sortierung
  - Per VBA einfach steuerbar
- Stabil, Bug-arm, Sicherheit, Gewissheit