Koordination innerhalb großer APEX Projekte

Referent:
Oliver Lemm,
Oracle Senior Berater,
MT AG, Ratingen

managing technology

MT AG Key Facts

MT AG MANAGING TECHNOLOGY – ENABLING THE ADAPTIVE ENTERPRISE

- Gründung 1994
- Inhabergeführte AG:
  Aktienkapital 1.500.000 €
- Hauptsitz Ratingen;
  Niederlassung Dortmund, Frankfurt
- Mitarbeiter:
  > 200 Festangestellte
  > 65 Freie Mitarbeiter

- Full-Service-Dienstleistung für alle Phasen des Software-Lifecycle
- Herstellerunabhängige Expertise in den marktführenden Technologien wie Oracle, IBM, Microsoft, SAP und OpenSource
- Themen- und Lösungs-Know-how in den Kerndisziplinen des Adaptive Enterprise

Agenda

- Standard APEX Projekte
  - Ticketsystem & Versionierung
- APEX Großprojekte
  - Team und Rollen
  - Konzept
  - Vorgaben/Rahmenbedingungen
  - Vorbereitungen
  - Transparenz
  - Ticketsystem
  - Versionierung
- Fazit

Standard APEX Projekte

- 80-90% der Projekte mit 1-3 Personen
- Alle Aufgaben liegen bei jeder Person
  - Datenbankänderungen, Seitenentwicklung, JavaScript, Querschnittsfunktionen, Anwendungslogik
- Wissensaustausch der Entwickler mit Auftraggeber/Fachbereich
- Organisation und Transparenz zwischen Entwicklern sind dabei nicht sehr komplex
- Rapid Application Development (RAD) steht dabei im Mittelpunkt

Ticketsystem & Versionierung

- Ticketsystem
  - Werden nicht zwangsläufig eingesetzt
  - Helfen aber auch bei einem einzigen Entwickler
    - Definition von Meilensteine & Hinterlegung von Bugs vom Auftraggeber
    - Nachvollziehbarkeit und Transparenz

- Versionierung
  - Installations- & Migrationsskripte
  - Hinterlegung von Stammdaten
  - Bereitstellung der Objekte für verschiedenen Umgebungen
  - Label / tags => Versionszuordnung

Agenda

- Standard APEX Projekte
  - Ticketsystem & Versionierung
- APEX Großprojekte
  - Team und Rollen
  - Konzept
  - Vorgaben/Rahmenbedingungen
  - Vorbereitungen
  - Transparenz
  - Das Ticketsystem
  - Versionierung
- Fazit

APEX Großprojekte

- Ab einer Anzahl von ca. 10 Personen die an der Entwicklung beteiligt sind
  - Unterschiedliches fachliches Wissen in APEX, PL/SQL, JavaScript, etc.
- Größe eines Projekts definiert sich nicht nur über die Anzahl Seiten
  - Größe wird über die Komplexität der Anwendungslogik, Schnittstellen, Abweichung von Standards beeinflusst
- Wissen der Anwendungslogik verteilt sich auf mehrere / verschiedene Personen/Entwickler
- Austausch mit Auftraggeber nur mit einem Teil der Personen/Entwickler

Team & Rollen

- Abhängig vom fachlichen Wissen der Entwickler und Umfang der verschiedenen Aufgaben müssen innerhalb des Projekts Rollen definiert werden
  - Projektleiter
    - Steuert das Projekt hinsichtlich der Ziele, Inhalte und Termine
    - Trifft keine technischen Entscheidungen
    - Ist nicht an der Entwicklung beteiligt
    - Sorgt für Transparenz und Motivation
    - Koordiniert die personelle Zuordnung und die Bereitstellung aller Hard & Softwareressourcen für die Entwicklung

Team & Rollen

- Architekt
  - Entwicklung des technischen Umsetzungskonzepts
  - Technisches und (falls möglich) fachliches Gesamtverständnis der Anwendung
  - Beurteilt Änderungen & Seiteneffekte
  - Abschätzung von Aufwänden
- Datenbankmodellierer
  - Struktur & Realisierung des Datenmodells
  - Rollout und Migrationsskripte
  - Stammdatenbereitstellung
  - Bereitstellung der verschiedenen Umgebungen

Team & Rollen

- APEX Entwickler
  - Entwicklung der APEX Oberfläche
  - Abbildung von jeglicher seitenspezifischer Sonderlogik
    - Defaults, Validierungen, Ausnahmen
- Datenbankentwickler
  - Allgemeine PL/SQL Logik
    - Schnittstellen, Querschnittsfunktionen, Testfunktionen
- Qualitätssicherung – QS
  - Testet die Anwendung auf Basis der Konzepte, Vorgaben und gemeldeter Bugs

Team & Rollen – Die Zusammenstellung

- Bei ca. 10 Personen kann die Zusammenstellung wie folgt aussehen
  - 1 Projektleiter
  - 1 Architekt
  - 2 Datenbankmodellierer
  - 1 QS
  - 3 APEX-Entwickler
  - 2 Datenbankentwickler

- Obige Zusammenstellung basiert auf einer Anwendung mit ca. 100 Masken, 400 Tabellen und einer großen Anzahl von Sonderlogiken und Querschnittsfunktionen
- Abweichungen in verschiedenen Bereichen (Validierungen, Laden & Speichern, Tabular Forms) von den APEX-Standards.

Konzept

- Alle Arbeiten mit den verschiedenen Konzepten werden von 2-3 Personen durchgeführt.
  - Aufwandsschätzung, Art der Umsetzung, Aufteilung der Inhalte
- Fachkonzept vom Auftraggeber zur Verfügung gestellt
  - Analyse des Fachkonzepts hinsichtlich
    - der Prozesse
    - der APEX Standards und technischen Möglichkeiten
    - der Rahmenbedingungen (Zeit, Geld und Entwicklerverfügbarkeit)
    - Aufteilungsmöglichkeiten in einzelne Pakete

Konzept

- Entwicklung des Umsetzungskonzept
  - Definition der Pakete & Schnittstellen zwischen den Paketen
    - Unterteilung in
      - technisch gleiche/ähnliche
      - Fachlich zusammenhängende Seiten
  - Fokus auf APEX Umsetzbarkeit
    - Standard vs. Eigenentwicklung
  - UML/ER-Definition der Relationen
  - Abbildung der Relationen bezogen auf die APEX Seiten

- Analyse des Fachkonzepts und Entwicklung des Umsetzungskonzepts sollten ca. 20% des Gesamtaufwands im Projekt repräsentieren

Vorgaben

- Eigene Vorgaben für
  - Datenbankmodellierung
  - Datenbankentwicklung
  - APEX-Entwicklung

- Vorgaben von APEX-Packages können sich von allgemeinen PL/SQL Packages unterscheiden

- Datenbankmodellierung
  - 3-4 stelligen Tabellenkürzel für Spalten, Trigger, Sequenzen verwenden
  - Möglichst restriktiv (Constraints, NOT NULL Spalten, …)
  - Tabellen und Spaltenänderungen nur vom Datenbankmodellierer

Vorgaben

- APEX-Entwicklung
  - Benutzung von Validierungen, Dynamic Actions und weiteren Mechaniken die allgemein definiert sind
  - Zwischenschicht zwischen Tabellen und allgemeinen Packages und den Seiten (Kapselung über Views und Seitenpackages)
    - Abhängigkeiten von Seiten zu Tabellen dadurch in der DB sichtbar
    - Invalide Objekte zeigen sofort die nicht lauffähigen Seiten
  - Browser
    - Version und verschiedene Browser können stark unterschiedlich im JavaScript reagieren.
  - Was darf nicht eingesetzt werden (konkurrierende Libraries, Templates die nicht angepasst wurden, …)

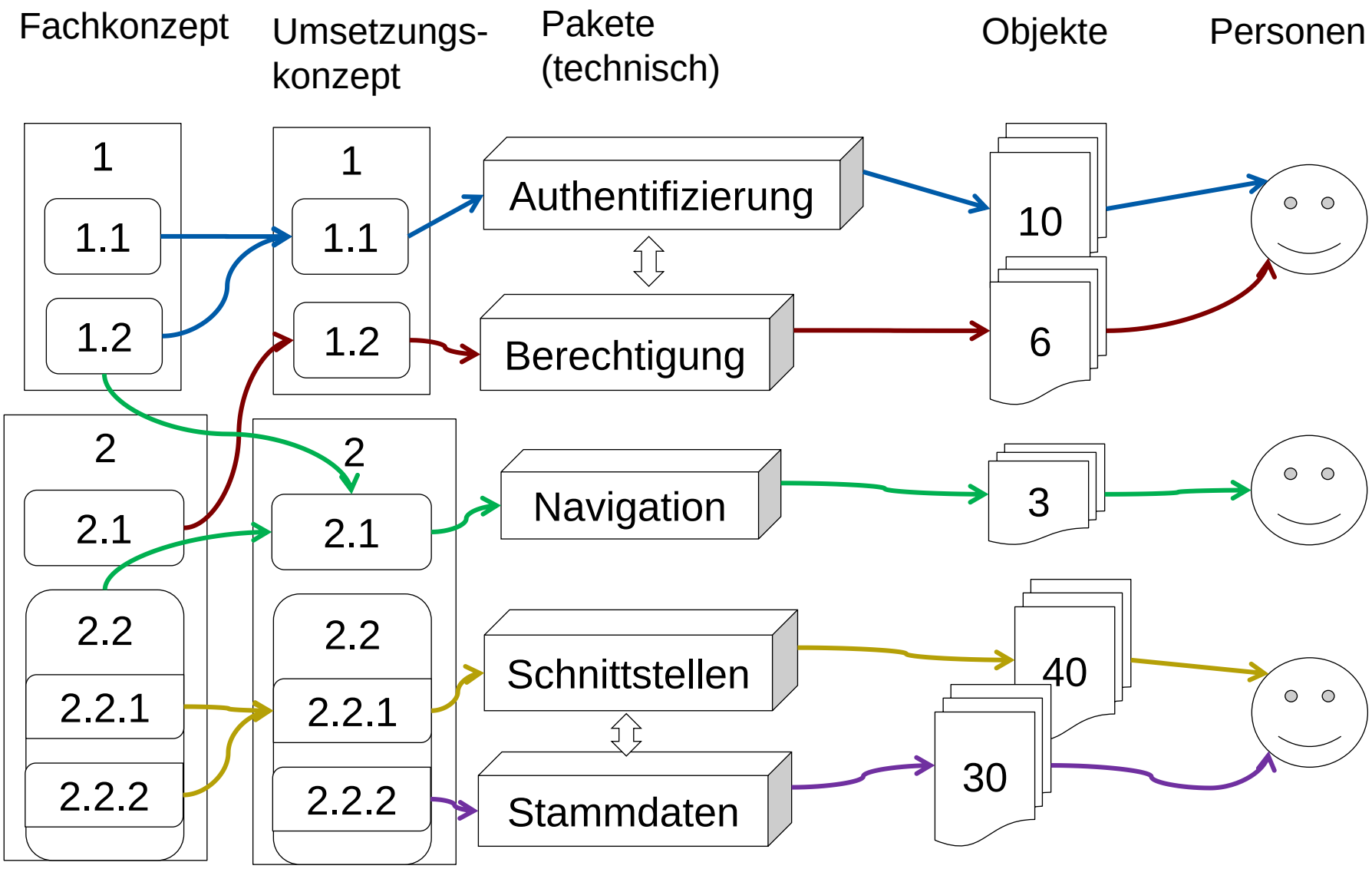
Vorbereitungen - Entwickler

- Bevor verschiedene Entwickler arbeiten müssen Werkzeuge, Prozesse, Systeme und Vorgaben definiert werden
  - Aufnahme aller Tools für
    - Modellierung
    - Versionierung
    - PL/SQL Entwicklung
    - Datenbankentwicklung
  - Speicherung und Versionierung muss kompatibel sein
    - Package & Spec zusammen
    - Sonderzeichen, Codeformatierung, Schemabezeichner
  - Zentrale Definition der Einstellungen pro Tool

Vorbereitungen - System

- Einrichtung der Systeme
  - Entwicklung, Test und Produktivsystem (ggf. weitere)
    - Dimensionierung
    - Version der Datenbank, APEX & Gateway
  - Versionierungssystem
  - Ticketsystem
  - Wiki
- Zentrale Bereitstellung aller Links
  - APEX Workspaces, Anwendungslinks, Versionierung, Ticketsystem, Wiki, Filesystem
- APEX Theme und Defaulteinstellungen

Transparenz



Transparenz

- Innerhalb des Projekts muss zu jedem Zeitpunkt jede Person wissen, wer für welches Objekt der Anwendung verantwortlich ist
  - Ein Objekt ist dabei eine APEX-Seite, ein Theme, jegliches Datenbankobjekt oder statische Dateien
  - Ein Objekt ist einem fachlichen Thema/Paket zugeordnet
  - Ein Thema/Paket ist eine APEX-Seite, eine Schnittstelle oder eine allgemeine Funktion wie Autorisierung, Authentifzierung, Navigation etc.
- Die Zuordnung zu Objekten und Themen/Paketen ist für einen Zeitraum zu definieren
- Der Zusammenhang zwischen Fachkonzept, Umsetzungskonzept, Thema/Paket und Objekt ist maßgeblich für den Erfolg des Projekts verantwortlich

Das Ticketsystem

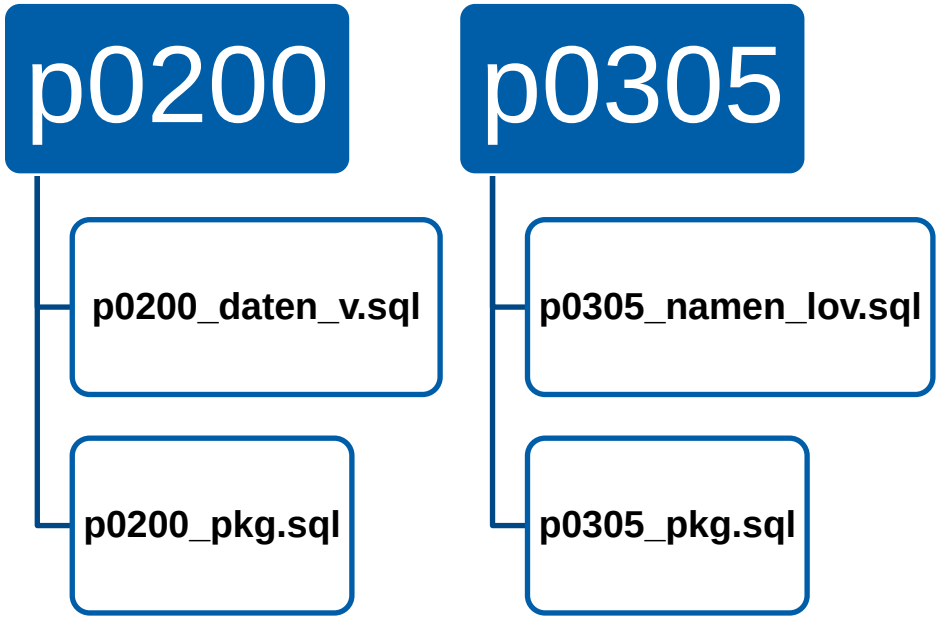
- Das Ticketsystem stellt die zentrale Schnittstelle zwischen allen Personen im Projekt dar
- Transparenz im Ticketsystem
  - Das Ticketsystem muss die Objekte und Themen/Pakete der Entwicklung abbilden
  - Es sollte den Bezug zum Fachkonzept und Umsetzungskonzept beinhalten
- Es muss ersichtlich sein, zu welcher Version/Auslieferung, welche Änderungen von wem durchgeführt werden.
- Jedes Objekt/Thema muss initial als „Hauptticket“ erstellt werden
- Es muss eine Zuordnung zu den Entwicklern, zu mindestens einem „Hauptticket“ bestehen.
- Es sollte eine Aufwandsschätzung und der tatsächliche Aufwand aufgenommen werden.

Das Ticketsystem

Allgemein	Eigenschaft	Ausprägungen
Person	Ersteller, Bearbeiter, QS, Gemeldet Durch	Liste der Mitarbeiter bzw. Kontaktpersonen
Version	Produktiv-, Ziel-, Geschlossen	Liste der Versionsnummern
Zusammenhang	Fachkonzept, Umsetzungskonzept, Paket, Objekt	Auswahllisten der Kapitel, bzw. Pakete und Objekte
	Gewichtung	
	Kategorie	Bug, Feature, ChangeRequest
Aufwand	Schätzung, Realisierung	0,25 Schritte in PT
Datum	Eingetragen, letzte Aktualisierung	Datumsangaben
	Dateien	Auszug Konzept, Screenshots, LOGS, Historische Daten

Versionierung

- Sobald mehrere Personen an der Anwendung arbeiten sind Locks auf den versionierten Dateien nötig
- Die Struktur/Kapselung der Objekte pro Seite sollte im Versionierungssystem abgebildet werden
  - Locking einfacher und schneller möglich
  - Alle Objekte zu einer Seite ersichtlich



- Verwendung von Properties/Keywords wie $author$, $date$, $revision$ etc.
  - Bezug zum Ticketsystem/Ticket über Property möglich

Versionierung

- Ablegen der Skripte muss vollständige Auslieferung und Updates (inklusive Datenmigration) beachten
  - Für Update/Migration ein Script mit DDL-Änderungen und Datenmigration in einer Datei ablegen
  - Weitere Möglichkeit: Immer eine Vollauslieferung durchführen und die Daten in einem Migrationsschema per XML Speichern und nach dem Update zurückführen.
    - Dabei ist kein Migrationsscript pro Änderung nötig und es werden keine „Alter“-Scripte für Tabellen benötigt die mit Daten umgehen müssen.
- Automatische Genierung eines Installationsscript über Tags-Ordner möglich
- Versionierung der Dateien über Branches oder Tags

Versionierung – Branches

- Jeder Entwickler erstellt einen eigenen „Zweig“ einer Datei und führt den Branch nach Fertigstellung in den /trunk zurück
- Probleme
  - Zum Testen bzw. Generierung einer Umgebung muss genau spezifiziert werden welche Branches benutzt werden
  - Die Anwendung muss mit „gebrancht“ werden
  - Abhängigkeiten zu anderen Objekten schwierig
  - Parallele Bearbeitung von mehreren Personen möglich aber nicht sichtbar
  - Pro Entwickler eine Entwicklungsumgebung nötig
- Vorteile
  - Änderungen über einen Versionszyklus hinweg sind möglich

Versionierung - Tags

- Alle Änderungen werden auf dem trunk durchgeführt.
- Bei Zusammenstellung einer Version werden alle zugehörigen Daten „getagged“
- Probleme
  - Dateien müssen gelockt werden
  - Gleichzeitiges Arbeiten an einer Datei nicht möglich
  - Änderungen für eine zukünftige Version schwer handhabbar
- Vorteile
  - Ausrollen einer Testversion einfach möglich (Tags-Ordner)
  - Koordination zwischen Entwickler einfacher => Durch Entwicklung auf einer Datei kann nicht aneinander vorbei entwickelt werden
  - Direkte Auswirkung des Packages in der Datenbank sichtbar

Versionierung – APEX Anwendung

- Versionierung der Anwendung als eine Datei
  - Änderungen in der Anwendung nur für nächste Version möglich
  - Ausrollen einzelner Seiten nicht möglich
- Bei Seitenexport und Weiterentwicklung sind folgende Punkte zu beachten
  - Änderung von LOV‘s, Navigation, Templates bzw. allgemein Shared Components und Änderungen auf Seite 0
  - Links von anderen Seiten
  - Import in andere Workspaces => Security ID unterschiedlich
  - Möglichkeit der Änderung einer einzelnen Seite

Fazit

- Wichtige Punkte vor allem in großen Projekten
  - Transparenz
  - Wissenstransfer und Aufgabenverteilung
  - Vorgaben & Vorbereitungen
- APEX eignet sich sehr gut für größere Projekte
  - Entweder mittels APEX-Standards
  - Oder über entsprechend genau definierte Prozesse und Abbildung der kompletten Logik in der Datenbank (nicht in APEX Seiten, nicht im JavaScript)
  - RAD – Ansatz in Großprojekten nur eingeschränkt möglich

Vielen Dank!

?!

MT AG managing technology I Balcke-Dürr-Allee 9 I 40882 Ratingen
Tel. +49 (0) 2102 309 61-0 I info@mt-ag.com I www.mt-ag.com

managing technology