

ORACLE®

# Der Oracle Application Express Entwicklungsprozess

Marc Sewtz  
Senior Software Development Manager  
Oracle Application Express

# Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

# Marc Sewtz

Senior Software Development Manager  
Oracle Application Express / Database Tools



- Mitarbeiter bei Oracle seit 1998
- Zunächst Oracle Consulting in Hamburg und New York
- Seit 2002 im Oracle Application Express Entwicklungsteam
  
- Twitter: @msewtz
- APEX on Twitter: #ORCLAPEX
- Blog: <http://marcsewtz.blogspot.com>

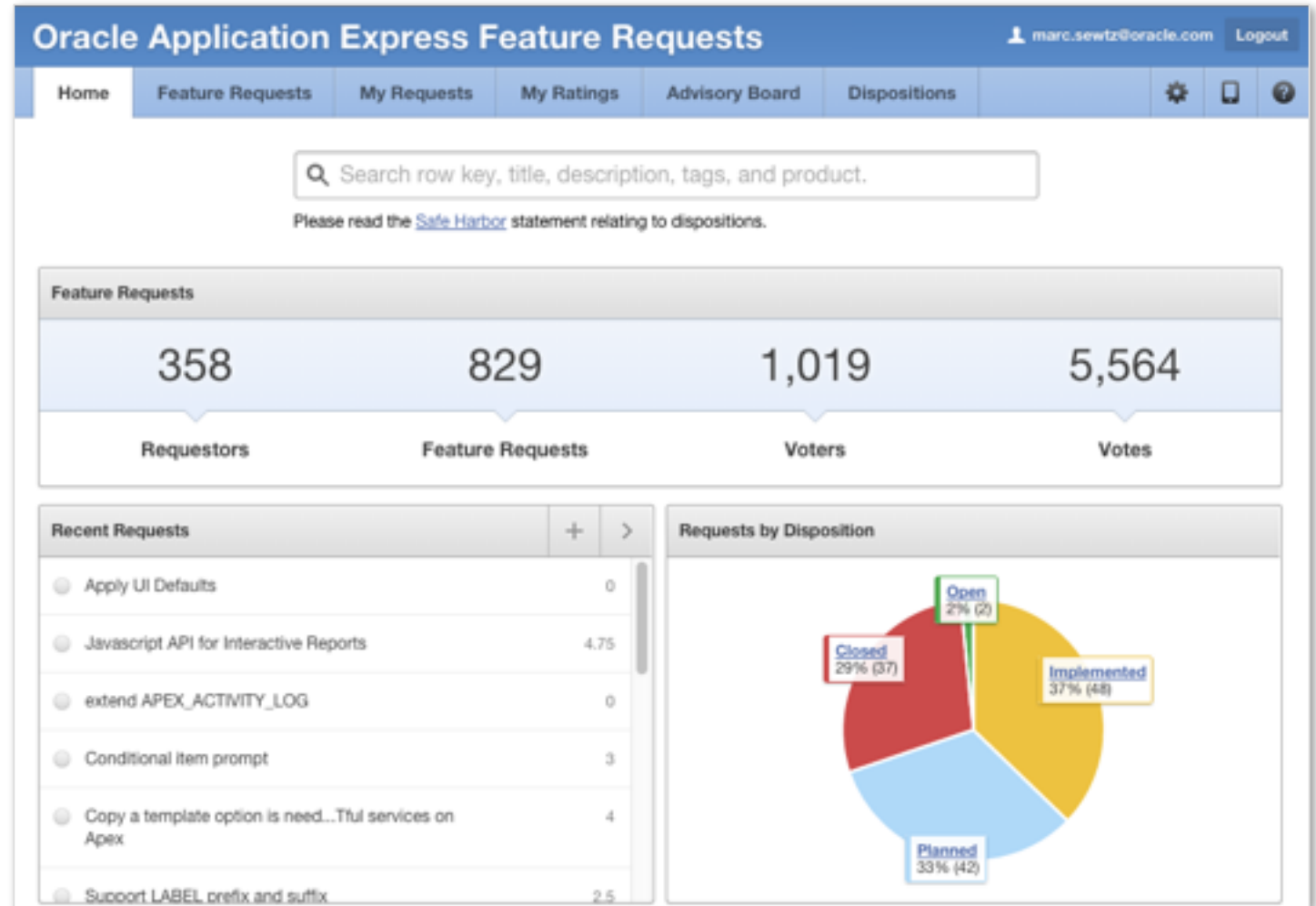
- 1** Feature Planung & Projekt Management
- 2** Entwicklung & Eingesetzte Werkzeuge
- 3** Qualitätssicherung und Auslieferung

# Feature Planung & Projekt Management

# Feature Planung

## Feature Request App

- APEX Community ist aktiv an der Planung neuer Features beteiligt
- Feature Advisory Board hat regelmäßige Review Meetings und Oracle an der Priorisierung
- FAB Meeting auf KScope



<http://apex.oracle.com/vote>

# Feature Planung

## Anforderungen die sich aus der Zusammenarbeit mit anderen Produkten ergeben

- Cloud
  - Database Schema Service und DBaaS unterstützen APEX
- Oracle Store
  - Mit APEX entwickelt
  - 150 Länder, 30 Sprachen
  - Schnittstelle zu über 12 back-office Systemen wie z.B. Manufacturing & Distribution, Marketing, EBS, etc.
- Packaged Applications
  - Theme & Theme Style Erweiterungen, Charts, APIs, etc





# Feature Planung

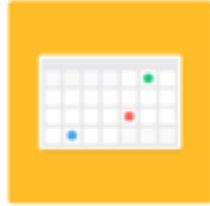
## Technologie Trends

- Integration moderner Frameworks und Libraries
  - JavaScript, AJAX, JSON, jQuery, HTML5, CSS3
  - jQuery Mobile, CodeMirror, Font-Awesome, Less
  - Menu Widget, Tree Widget, Splitter Widget, IconList Widget



# Projekt Management

## Team Development



Milestones



Features



To Dos



Bugs



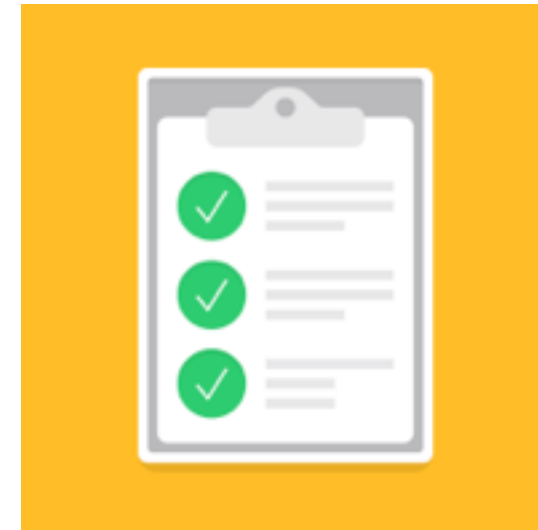
Feedback

- Projekt Management Werkzeug für die Entwicklung von APEX Apps
- Milestones, Features, ToDos, Bugs, Feedback
- Apps mit aktivierter Feedback-Funktion erlauben Endbenutzern direkt Feedback an die Entwicklung zu schicken
- Session Information automatisch im Feedback enthalten
- Anlegen neuer Features, ToDos oder Bugs basierend auf Feedback

# Projekt Management

## Release Management

- Track Milestones and Feature Completion
- Bug Triaging
- Documentation Milestones
- Translation Drops
- Globalization Testing
- Accessibility Testing
- Security Analysis
- Approvals for Open Source Libraries
- Early Adopters, User Acceptance Testing



# Demo

# Entwicklung und eingesetzte Werkzeuge

# Feature Entwicklung

## Beispiel-Szenario

- Feature Beschreibung (fiktives 5.1 Feature):
  - ① Template Option Groups sollen mit einem neue Attribute „Required“ erweitert werden, mit diesem Attribute wird definiert, ob die Auswahl einer Template Option erforderlich ist
  - ② Das Universal Theme soll aktualisiert werden: das Standard Region Template soll eine neue „Accent 6“ Option erhalten und Accents sollen als „Required“ definiert werden



# Feature Entwicklung

## Beispiel-Szenario – neues Attribute

- DDL Änderungen (Tabellen / Views)
  - Neue Spalte „required“ für Tabelle „wwv\_flow\_template\_opt\_groups“
  - Update des „apex\_appl\_template\_opt\_groups“ Views
- eingesetzte Werkzeuge:
  - Subversion Client (z.B. Cornerstone): auschecken/einchecken der DDL Skripte
  - Oracle SQL Developer: manuelles „Alter Table“
  - Code Editor (z.B. Sublime): editieren der DDL Skripte

# Feature Entwicklung

## Beispiel-Szenario – neues Attribute

- PL/SQL Änderungen
  - Erweiterung des APEX Exports (GENAPI): dem Export der Template Option Groups muss das neue Attribute hinzugefügt werden
  - Erweiterung des APEX Imports (API): Import API muss optional das neue Attribute akzeptieren, bzw. wissen was zu tun ist, wenn ältere Apps importiert werden
  - Aktualisierung the Copy und Translation APIs für Anwendungen
  - Aktualisierung der Copy APIs für einzelne Seiten
  - Aktualisierung der Wizard APIs wenn Auswahl im Wizard unterstützt ist
  - Aktualisierung der Theme / Template / Shared Components APIs – in diesem Fall Kopieren der mit dem Theme assoziierten Template Option Groups



# Feature Entwicklung

## Beispiel-Szenario – neues Attribute



- PL/SQL Änderungen – eingesetzte Werkzeuge
  - Subversion Client (z.B. Cornerstone): auschecken / einchecken der PL/SQL Packages
  - Code Editor (z.B. Sublime): editieren der PL/SQL Packages
  - Oracle SQL Developer oder SQL Plus: Kompilieren der PL/SQL Packages
  - APEX Builder UI: Testen des Export / Imports und Kopieren von Apps und Seiten

# Feature Entwicklung

## Beispiel-Szenario – neues Attribute

- UI Änderungen
  - Aktualisierung der APEX IDE(Application Builder)
  - Template Option Groups können als Teil der Theme Definition editiert werden (Shared Components > Themes > Template Option Groups)
  - Template Option Group Edit Dialog neues Attribute „Required“ hinzufügen
- eingesetzte Werkzeuge:
  - APEX Builder UI: Die APEX UI selbst wird mit APEX entwickelt, alle APEX UI Änderungen werden daher direkt in APEX vorgenommen (Editieren der App 4000)

# Feature Entwicklung

## Beispiel-Szenario – neues Attribute

- Aktualisierung des “Template Option” Native Item Plug-Ins
  - Template Option Auswahlfeld erlaubt Auswahl von Template Option für Seiten, Regionen, Reports, Menus, Breadcrumbs, Felder und Buttons sowie den zugehörigen Templates
  - Template Option Auswahlfeld als native Plug-In realisiert
  - Plug-In dahingehend erweitern, dass erforderliche Template Optionen entsprechend markiert werden und ein Validierungsfehler angezeigt wird, wenn keine Option ausgewählt wurde

# Feature Entwicklung

## Beispiel-Szenario – neue „required“ Template Option

- Änderung der Template Option Group „Accent“
  - Editieren des Universal Theme Masters
  - Unter Shared Components > Themes > Template Option Groups, die Gruppe „Accent“ zum Editieren öffnen
  - Neues Attribute „Required“ aktivieren
- eingesetzte Werkzeuge:
  - APEX Builder UI: Die Universal Theme Master App sowie all Packaged Apps werden in einem separaten Workspace auf dem APEX Development-Server entwickelt

# Feature Entwicklung

## Beispiel-Szenario – neue „required“ Template Option

- Änderung des „Standard“ Region Templates
  - Im Universal Theme Master, das „Standard“ Region Template zum editieren öffnen
  - Unter Template Optionen, eine neue Template Option „Accent“ 6 anlegen, mit der folgenden CSS Class: t-Region--accent6

# Feature Entwicklung

## Beispiel-Szenario – neue „required“ Template Option

- Hinzufügen einer neuen CSS Class
  - CSS Dateien in: /apex/images/themes/theme\_42/1.0/css
  - In Development als SASS definiert und modularisiert (CSS Preprocessor)
- eingesetzte Werkzeuge:
  - Sublime (Editieren von CSS/SASS Files)
  - SASS – Syntactically Awesome Style Sheets / CSS Preprocessor
  - Grunt – JavaScript Task Runner, automatisiert das Erzeugen von CSS Files, Less Files sowie Minification und Processing von Image Ressourcen
  - GruntIcon – Transformiert SVG Grafiken in CSS und PNG Fallbacks

# Source Control System

## Subversion

- Versions-Verwaltung von Dateien und Verzeichnissen
- Branches für verschiedene APEX Versionen und Patch-Sets
- Alle APEX Komponenten in SVN verwaltet:
  - DDL und PL/SQL
  - JavaScript, CSS, Images, Third-Party Libraries
  - Themes
  - APEX Application Exports (4xxx Apps / nächtlicher Export)
  - Packaged Application Exports (nächtlicher Export)



# Demo



# Qualitätssicherung und Auslieferung

# Qualitätssicherung

## Wie wird APEX getestet?

- Laufendes Testen während der Entwicklung
- QA – Neue Features, Regression Testing (Selenium / Browser Automation)
- Globalization Testing
- Linguistik Checklist
- Accessibility Testing
- Security Analysis
- Security Checklist

# Accessibility Testing

## Barrierefreiheit

- Alle Oracle Produkte und Services müssen entsprechend Oracle's Accessibility Guidelines getestet werden (OAG 2.0)
- Jedes Produkt, das von Kunden verwendet werden kann, muss einen VPAT (Voluntary Product Accessibility Template) haben
- Ein VPAT besagt, zu welchem Grad ein Produkt der Section 508 des U.S. Federal Rehabilitation Acts genügt
- Produkte, die nicht den minimal Anforderungen genügen, müssen eine Roadmap veröffentlichen, die besagt, wann die Probleme behoben werden

# Accessibility Testing

## Barrierefreiheit

- Keyboard-only Testing
  - Ist es möglich, eine APEX Anwendung ausschließlich mit dem Keyboard zu bedienen?
- JAWs Testing
  - Job Access With Speech - Screen Reader hergestellt von Freedom Scientific
  - Ist es möglich APEX per Screen Reader und Spracheingabe zu bedienen?
- OGHAG/OAG Helper Toolbar Testing
  - Valid HTML, ausreichendes Markup für Tabelle, Form Felder Images, etc
  - Verständliche Seitenstruktur
  - Ausreichender Farbkontrast

# Oracle Application Express Build Process

## Kontinuierlichen Integration mit Hudson

- Nächtllicher Build:
  - Export aller internen APEX Apps (Builder, SQL WS, Team Dev, ...)
  - Export aller Packaged Apps und Theme Apps
  - Subversion Check-In der Export Files
  - Ausführen der Grunt-Skripte
    - Transformieren und Minimizing von SASS in CSS und Less
    - Minimizing und Uglifying von JavaScript
  - Test Build aller auf Staging Instanz von aktuellem Stand in Subversion
  - Error Log protokolliert mögliche Probleme für die Entwickler

# Auslieferung von APEX Apps in die Produktion

## Konsistente Workspace- und Anwendungs-IDs

- Manuelle Auswahl der ID beim Anlegen eines Workspaces
- Gleiche Workspace ID auf allen Instanzen (Entwicklung, QA, Produktion)
- Export / Import des Workspaces in Workspace- oder Instanz-Administration
- „Platzhalter“ Anwendungen auf allen Instanzen
- Anwendungs-Import immer mit der gleichen Anwendungs-ID

# Auslieferung von APEX Apps in die Produktion

## Konsistente Workspace- und Anwendungs-IDs

- Vereinfacht die Replikation und Aktualisierung von APEX Anwendungen und Komponenten über verschiedene APEX Instanzen
- Automatisierte Installation und Aktualisierung von APEX Anwendungen und Komponenten via SQL Plus
- Ermöglicht Import und Aktualisierung einzelner Seiten und Komponenten

# Auslieferung von APEX Apps in die Produktion

## Beispiel-Workflow für die Auslieferung von Anwendungen und Komponenten

- ① Entwickler exportiert die Anwendung, optional inkl. Supporting Objects
- ② APEX generiert Export SQL Script
- ③ Entwickler checkt SQL Script in Source Control System ein
- ④ Entwickler übergibt „Run Sheet“ an DBA
- ⑤ DBA lädt erforderliche Dateien aus Source Control System
- ⑥ DBA führt SQL Script auf QA und Produktion-System aus



# Auslieferung von APEX Apps in die Produktion

## Verwendung von Build Options

- Build Options erlauben es, Teile der Anwendung abhängig von bestimmten Bedingungen anzuzeigen oder auszublenden
- Build Optionen können für die meisten Komponenten ausgewählt werden
- Komponenten können dann z.B. auf einer Entwicklungsinstanz angezeigt werden, in der Produktion aber ausgeblendet sein
- Damit ist es möglich, neue Features selektiv in die Produktions zu bringen

# Auslieferung von APEX Apps in die Produktion

## Komponenten Export

- APEX Anwendungen können entweder komplett exportiert werden oder es können einzelne Komponenten selektiv exportiert werden
- Der Export einer einzelnen Seite erlaubt es z.B. nur die Änderungen auf dieser Seite in die Produktion zu bringen
- Voraussetzung ist, das konsistente Workspace und Anwendungs-IDs verwendet werden

# Demo

# Q & A

# **Hardware and Software Engineered to Work Together**

ORACLE®