Koordination innerhalb großer APEX Projekte

Oliver Lemm
MT AG
Ratingen

Schlüsselworte:

APEX, Versionierung, Ticketsystem, Subversion, Koordination

Einleitung

Im Vortrag wird dargestellt, was für Herausforderungen bei der Entwicklung von APEX Anwendungen mit mehreren Entwicklern auftreten können. Es werden Vorschläge zum Umgang mit der Versionierung und der Koordination über ein Ticketsystem dargestellt.
Zusätzlich wird die Kapselung der Logik im Bereich PL/SQL angesprochen und Anregungen gegeben wie die Kapselung in APEX vorgenommen werden kann.
Da im Vortrag die Koordination in großen APEX Projekten im Mittelpunkt steht, werden einige sehr wichtige Ansätze in der Entwicklung, Planung, und dem Rollout nur am Rande erwähnt.

Kleine APEX Projekte

Entwicklung

Ein Großteil von heutigen APEX Projekten wird von Einzelpersonen oder Teams von bis zu 3 Leuten durchgeführt. Dabei liegen oft mehrere, bis alle Aufgaben bei einer Person.
Auch das Wissen über das Gesamtsystem kann jedem Entwickler vollständig zur Verfügung gestellt werden. In diesen Projekten findet der Wissensaustausch vom Fachbereich/Auftraggeber zum Entwickler in der Regel mit allen Entwicklern statt und die Abstimmung der Aufgaben und Änderungen kann direkt mit allen Personen besprochen werden.
Meist organisiert man diese Projekte so, dass sowohl die Änderungen in der Datenbank, als auch die zugehörigen Änderungen in der Anwendung in einem Schritt und von einer Person durchgeführt werden.

Ticketsystem

In solchen kleineren Projekten werden oft keine Ticketsysteme eingesetzt.
Dies kann jedoch schon bei einer einzelnen Person hilfreich sein um Änderungen/Bugs oder Versionen/Meilensteine zu definieren. Ein Ticketsystem bietet jederzeit die Möglichkeit transparent noch offene Punkte in der Entwicklung aufzuzeigen oder Bugs festzuhalten.

Versionierung

Unabhängig von der Größe in APEX Projekten sollte eine Versionierung der Datenbankobjekte, Anwendung, Stammdaten und der statischen Dateien (Grafiken, JavaScript und CSS) stattfinden.
Hierbei muss man als Einzelperson bei jeder Änderung die durchgeführt wurde die entsprechenden Dateien einchecken.
Am Ende der Entwicklung bzw. eines Patches wird dann ein „Label“/“Tag“ über den letzten Stand definiert worüber die Dateien identifiziert werden die auf ein Test/Produktivsystem gespielt werden.

APEX Großprojekte

Ab einer Anzahl von ca. 10 Entwicklern kann man aus meiner Sicht von einem APEX Großprojekt sprechen. Hierbei sind neben den fachlichen Fertigkeiten im Bereich APEX und Datenbankentwicklung eine Reihe weiterer Punkte ausschlaggebend, welche für den Erfolg eines Projekts maßgebend sind. Einige dieser Punkte sind unabhängig von APEX Projekten gebräuchlich, aber besitzen innerhalb von APEX Projekten andere Schwerpunkte. Wichtig ist weiterhin, dass nicht die Anzahl der Seiten eine Aussage über die Größe eines APEX Projekts machen, sondern allgemein die Komplexität der Anwendung und enthaltenen Logik.

Das Team

Folgende Rollen sind in größeren APEX Projekten auf ein oder mehrere Personen abzubilden:

Der Projektleiter

Der Projektleiter übernimmt die Steuerung des Projekts hinsichtlich der Ziele, Inhalte und Termine des Projekts. Er übernimmt keinerlei Entwicklungstätigkeit und trifft keine technischen Entscheidungen. Er koordiniert die Mitarbeiter im Team und sorgt für Transparenz im Projekt.
Gerade in einem großen Projekt ist es wichtig, dass er zu jederzeit einen Überblick über die Aufgaben und Ziele im Projekt hat und dem Team genau sagen kann, wann welche Person mit welchen Aufgaben betreut ist.

Der Architekt

Der Architekt besitzt ein technisches und fachliches Gesamtverständnis der Anwendung. Er kann beurteilen, welche Änderungen an welcher Stelle Auswirkungen haben und ist in der frühen Phase des Projekts enorm wichtig für die Konzeption der Anwendung und des Datenbankmodells. Bei großen Datenbankmodellen (mehr als 200 Tabellen) kann man allein fürs Datenbankmodell einen einzelnen Architekten abstellen.

Der Datenbankmodellierer

Dieser kümmert sich um die Struktur und Realisierung des Datenmodells. Auch stellt er ER-Diagramme zur Verfügung und kümmert sich um die Stammdaten.
Zuletzt liegt ein Schwerpunkt in Auslieferungen, die ab einer bestimmten Projektphase meist inkrementell und vollständig sein müssen.

Der Datenbankentwickler

Dieser kümmert sich um zentrale Logik, welche nicht seitenspezifisch entwickelt wird. Berechtigung, Schnittstellenlogik und jegliche fachlich komplexe Logik die innerhalb der Datenbank abgebildet wird.

Der APEX Entwickler

Dieser kümmert sich um die Seitenentwicklung und der seitenspezifischen Logik. Evtl. nötige Änderungen an Tabellen oder an zentralen Funktionen koordiniert er mit den jeweiligen

Verantwortlichen Personen (Tabellen => Datenbankmodellierer, zentrale Funktionen => Datenbankentwickler, Sonderfälle/Ausnahmen => Architekt).

Die Qualitätssicherung

Die Personen testen jeden Sachverhalt der entwickelt wird. Neben der Anwendung werden Ihnen von den Datenbankentwicklern Testfunktionen/Unit Tests zur Verfügung gestellt, die Ihnen ermöglichen jeden Sachverhalt zu testen.
Die Qualitätssicherung arbeitet auf einem Testsystem.

Die Zusammenstellung

Bei einem Projekt mit einer Größe von 10 Personen wäre Zusammenstellung ca. folgendermaßen:
1 Projektleiter
1 Architekt
2 Datenbankmodellierer
1 QS
3 APEX-Entwickler
2 Datenbankentwickler

Die Zusammenstellung variiert natürlich stark von der Ausprägung der Logik und der Anzahl der Seiten. Bei einer Anwendung mit einem kleinen Datenbankmodell (keine 100 Tabellen) und einer sehr seitenspezifischen Logik muss die Anzahl der APEX-Entwickler höher sein und die Datenbankentwickler, sowie Datenbankmodellierer kleiner. Zusätzlich können Personen zum Controlling und zur Steuerung des Projekts hinzugezogen werden.

Das Konzept

Der Beginn eines Großprojekts basiert anfangs auf einer kleinen Anzahl von Personen (ca. 2-3) die basierend auf einem Fachkonzept ein technisches Umsetzungskonzept definieren.
Dabei sollte anfangs der Fokus auf der gesamten Struktur einer Anwendung und den Möglichkeiten der Unterteilung in kleinere Pakete liegen. Innerhalb von APEX Projekten sollte man darauf achten, dass diese Unterteilung so vorgenommen wird, dass einerseits das nötige Wissen für ein Paket als in sich geschlossen angesehen werden kann und wie dies technisch in APEX abgebildet werden kann.
Zum Beispiel kann in APEX keine beliebige Vererbung von Seiten stattfinden. Daher kann man entweder Seiten, welche in der gleichen Technik erstellt werden in einem Paket zusammenfassen, oder man fasst alle Seiten eines fachlichen Bereichs zusammen.
Aus meiner Erfahrung heraus, ist der fachliche Zusammenhang meist sinnvoller, da das Verständnis des APEX-Entwicklers und des Datenbankmodellierers beim Entwurf der Seiten und des Datenbankmodells ein größer ist.
Das Umsetzungskonzept sollte im Gesamtaufwand ca. mit 20% der Gesamtzeit eingerechnet werden.

Die Vorgaben

Im Bereich der Datenbankmodellierung, Datenbankentwicklung und APEX-Entwicklung sind jeweils genaue Vorgaben zu definieren. Diese können parallel zum Konzept erstellt werden. Hierbei kann man einerseits definierte Vorgaben aus früheren Projekten übernehmen, muss aber ggf. zusätzliche Vorgaben welche beim Kunden hinsichtlich der Datenbank und APEX gelten beachten.

In der Datenbankmodellierung sollte man hinsichtlich APEX immer mit möglichst restriktiven, normalisieren Datenbankmodellen arbeiten. Dies sorgt für eine hohe Unterstützung der APEX Standards und eine hohe Datenqualität. Man sollte auf technische Primärschlüssel achten, damit die Struktur der Daten nicht durch inhaltliche Änderungen beeinflusst werden.
Auch sollte man innerhalb der Datenbank auf Tabellen 3 bis 4 stellige Tabellenkürzel achten umso die Arbeit mit mehreren Tabellen transparenter zu gestalten.

Als weiteren wichtigen Punkt hat sich der Umgang mit einer Zwischenschicht zwischen Tabellen und zentralen Funktionen und den APEX Seiten herausgestellt. Man definiert für jede Seite eine oder mehrere Views/Packages, welche die Informationen/Logik enthalten, welche in der jeweiligen Seite benötigt werden. Hierdurch wird innerhalb der Datenbank jegliche Abhängigkeit von Datenbankobjekten mit einer APEX Seite ersichtlich und man kann ohne die Anwendung ausführen innerhalb der Datenbank erkennen, welche Seiten bei Änderungen oder invaliden Objekten betroffen sind. Dies stellt in Großprojekten einen enorm wichtigen Punkt dar, damit einem Blick in die jeweiligen Datenbankschemata klar ist ob eine APEX Seite lauffähig ist oder nicht.
Als zusätzliche Absicherung kann man alle Objekte innerhalb einer Seite in einer Tabelle festhalten um die Vollständigkeit der Objekte für die Anwendung zu gewährleisten.

Neben den Vorgaben, wie man Schritte umsetzt muss auch klar abgegrenzt werden, was nicht eingesetzt werden soll/darf. Sind Besonderheiten beim Browser, bei JavaScript bei APEX Objekten vorhanden die nicht unterstützt werden oder zu Seiteneffekten führen.
Besonders, wenn ein Standardverhalten von APEX nicht ausreicht oder der Kunde eine besondere Logik an einer Stelle der Anwendung benötigt muss man genau mit dem Kunden ermitteln, was das Ziel ist und ob ggf. eine andere Lösung die evtl. leichte Abstriche in der Bequemlichkeit bedeutet, aber dafür in der Wartung und Fehleranfälligkeit einen Vorteil darstellt.
Gerade in Großprojekten muss jegliche Abweichung vom Standard und nicht definierte Mechanik mit dem Architekten und weiteren Seitenentwicklern abgestimmt werden, da ansonsten verschiedene Stellen in der Anwendung unterschiedlich arbeiten. Am Schluss bedeutet dann für den Endbenutzer, dass er unterschiedlichste Seiten für ähnliche Sachverhalte wiederfindet.

Für jeden die drei Bereiche Datenbankmodellierung, Datenbankentwicklung und APEX-Entwicklung sollte jeweils ein Dokument bereitgestellt werden, welches alle jeweiligen Vorgaben enthält. Auch wenn ein Datenbankentwickler und ein APEX-Entwickler beide Packages/Views erstellen, so können die jeweiligen Vorgaben voneinander sinnvoll abweichen.

Die Vorbereitungen

Bevor die Entwicklung beginnt müssen die Systeme und Tools definiert werden mit welchen die Entwicklung durchgeführt wird.

Die Entwicklervorbereitungen

In einem großen Projekt passiert es nach meiner Erfahrung oft, dass verschiedene Personen den gleichen Code bearbeiten und wenn dann durch unterschiedliche Formatierungen im Code oder andere Einstellungen im Tool die Objekte unterschiedlich speichern, führt dies selbst bei Änderung einer Zeile dazu, dass das Package komplett unterschiedlich dargestellt wird und innerhalb der Versionierung als komplett unterschiedlich gespeichert wird.

Sind also Tools definiert, so müssen entsprechend die Einstellungen auf den verschiedenen Entwicklerrechnern entsprechend übernommen und abgestimmt werden. Dies kann man einerseits

über eine entsprechend Dokumentation pro Tool hinterlegen und so von Hand übernehmen, aber zusätzlich besitzen einige Tools Konfigurationsdateien, welche nur in das jeweilige Verzeichnis des Entwicklungstools kopiert werden können.

Je nach Qualifikation der Mitarbeiter muss neben dem Umgang mit dem Tool, ggf. eine fachliche Einführung stattfinden. Diese Einführung bzw. Vorgaben sind in einem großen Projekt enorm wichtig, da ansonsten bei der Versionierung oder Bearbeitung der Tickets und damit in der Kommunikation Daten verloren gehen bzw. nicht korrekt abgelegt werden.

Die Systemvorbereitungen

Enorm wichtig für größere Projekte ist die Bereitstellung der verschiedenen Systeme. Zur Entwicklung, zum Testen und zur Produktionsumgebung werden separate Systeme benötigt. Die Dimensionierung der Systeme hängt von der Komplexität der Funktion, den Daten, sowie den simultanen Benutzern ab. Auch kann es je nach Projekt nötig sein, weitere externe Systeme anzubinden. Für die Koordination der Entwickler und QS ist dabei wichtig eine zentrale Stelle/Webseite/Wiki zu besitzen, wo genau definiert ist, welche Umgebung mit welchen Benutzern/Daten und Links zu erreichen ist.
Sind bei der APEX Entwicklung, Anwendungslinks, Datenbankverbindungen, Art des Systems und Inhalt der Daten unklar, so führt dies zwangsläufig zu Fehlern und Problemen.
Auch sollte man es dem Kunden ermöglichen auf die Test- & Produktivumgebung zuzugreifen, um evtl. Sachverhalte vorzuführen oder Fehler, die auf weiteren Systemen vorgefallen sind darzustellen.

Für diese Struktur bietet sich eine Virtualisierung der Umgebung an, wobei in großen Projekten ggf. ein physisches System für Besonderheiten sinnvoll und nötig sein kann.

Innerhalb der APEX Umgebung sollte darauf geachtet werden, dass die Entwickler Zugriff auf alle Workspaces haben. Zusätzlich sollte man so viele Vorgaben wie möglich innerhalb des APEX Themes abbilden. Innerhalb des Umsetzungskonzepts sollten dabei möglichst viele Vorgaben für die APEX Objekte wie Items, Regionen, Seiten definiert werden.
Es ist sehr hilfreich entweder das aktuellste APEX Theme zu nehmen und zu „entschlacken“ oder eigenes Theme definieren. In beiden Fällen sollte man auch hier darauf achten zum Beispiel über eine Namenskonvention alle Objekte die benutzt werden dürfen mit einem Kürzel zu versehen. Sind Objekte noch nicht innerhalb des Projekts definiert so muss auch hier der Weg über den Architekten gehen, der dies dann mit dem APEX-Entwickler abstimmt.

Die Transparenz

Innerhalb des Projekts muss zu jedem Zeitpunkt zu jedem Objekt, was entwickelt wird ein primärer Ansprechpartner feststehen. Ein Objekt ist eine APEX-Seite, das APEX Theme, jegliches Datenbankobjekt oder statische Dateien wie JavaScript, Grafiken und CSS.
Ein Objekt wird dabei einem fachlichen Thema zugeordnet. Dies kann auf kleinster Ebene eine Seite sein, eine ganze Schnittstelle oder eine bestimmte Mechanik in der Anwendung. Diese Zuordnung sollte auf Basis eines Zeitraums stattfinden und es muss durch den Projektleiter gewährleistet sein, dass kein Objekt zu einem Zeitpunkt ohne eine Verantwortliche Person vorhanden ist.
Diese Zuordnung kann man über eine APEX Anwendung oder andere Planungstools unterstützen und müssen für jede Person jederzeit abrufbar sein.
Neben der Transparenz für die Entwicklung kann man zusätzlich die Kapitel aus dem Fachkonzept und Umsetzungskonzept in Bezug zu dem fachlichen Thema zuordnen. Dadurch erhält man vom einen

Zusammenhang zwischen den Inhalten des Fachkonzepts, über das Umsetzungskonzept, über die Thematik bis hin zum Datenbankobjekt.

Das Ticketsystem

Das Ticketsystem stellt in Großprojekte die zentrale Schnittstelle zwischen allen Personen im Projekt dar und trägt in richtiger Verwendung schon maßgeblich zur Transparenz bei. Die einzelnen Zuordnungen, welche im Bereich der Transparenz definiert sind, lassen sich oft im Ticketsystem repräsentieren. Dabei sollte man die Kapselung „fachliches Thema“ als Kategorie und als Unterkategorie „Objekt“ im Ticketsystem abbilden können. Zusätzlich sollte bei einem Ticketsystem die Möglichkeit bestehen Screenshots bzw. Dokumente mit anzuhängen.

APEX Team Development

Der Sachverhalt, dass man in APEX keine Dateien im internen Team Development anhängen kann und dass das Team Development primär mit der Seitenentwicklung verzahnt ist, schließen es leider für größere Projekte aus.
Das Team Development aus APEX kann man zur Rückmeldung von Bugs des Kunden benutzen. Der riesige Vorteil der integrierten Bugmeldung sind die umfangreichen Informationen von Items und Inhalten der Seite. Sind in der dortigen Umgebung oder ggf. auch schon in der Testumgebung Bugs angefallen, so kann man diese exportieren und die gemeldeten Bugs auf der Entwicklung importieren. Neben der Meldung über das Ticketsystem kann man so pro Bug auch die Werte der Anwendung bekommen und so den Bug einfacher und schneller nachstellen.

Die Struktur

Für jedes Objekt sollte bei der initialen Entwicklung ein „Hauptticket“ angelegt werden. Diese sollten über die Eigenschaft des „Haupttickets“ auch immer wieder zu identifizieren sein. Jeglicher Bug, jede Änderung die in der Entwicklungsphase auftritt wird dort vermerkt.
Jede Änderung nach der ersten Abschluss der initialen Entwicklung dieses Objekts sollte dann als Unterticket an dieses Ticket angehängt werden und je nach Sachverhalt (Bug, Feature, Datenbankänderung) etc. vermerkt werden.

Neben der Struktur der Tickets müssen die Tickets über Versionen, welche im Versionierungssystem als Label/Tag verwendet werden zugeordnet werden können.
Nur so ist es möglich schnell an Informationen und Änderungen in einem großen Projekt ranzukommen ohne das jede Person direkt gefragt werden muss.

Das Ticketsystem sollte weiter auch die Möglichkeit haben darzustellen, wann der Fehler, in welcher Version gemeldet wurde, für welche Version die Änderung geplant ist und bei vergangenen Tickets darstellen, wann der Fehler behoben wurde. Und nicht zuletzt sollte der Aufwand dargestellt werden, welcher geschätzt wurde und welcher Aufwand tatsächlich angefallen ist.
Gerade die Planung in einem großen Projekt bzgl. der einzusetzenden Personen, sowie der Aufwand den jeder Sachverhalt darstellt müssen abgebildet werden.

Die Versionierung

Neben den relevanten Punkten für die Versionierung, welche für kleinere Projekte gelten, sind für Großprojekte die Mechaniken wesentlich komplexer. Wird ein Datenbankobjekt geändert, so muss

man gewährleisten, dass keine parallele Entwicklung durch eine zweite Person stattfindet. Dies kann man innerhalb der Versionierung über „Locks“ erreichen. Hierbei sollte man darauf achten, dass man wenn möglich die Objekte innerhalb der Versionierung zum Beispiel pro Seite bündelt damit ein "Locking" auf allen Objekten der Seite stattfinden kann. Würde man an dieser Stelle keine Schicht zwischen den Seitenobjekten (Views/Packages) und den allgemeinen Objekten haben, würde man nicht direkt bei der Entwicklung und beim Sperren der Objekte merken, dass mehrere Entwickler ggf. am gleichen Objekt Änderungen vornehmen.

Bringt man nun eine zukünftig neue Version auf den Weg, muss jeder Entwickler selbst definieren können, ab welchem Zeitpunkt sein Package mit in eine Auslieferung/neue Version kommt. In anderen Programmiersprachen führt man an dieser Stelle oft den Weg eines "Branches" durch.
Dieser lässt sich in der APEX Anwendung aber nicht so einfach darstellen. Da die APEX Anwendung als Ganzes benötigt wird müsste entweder jeder Entwickler seine eigene Entwicklungsumgebung und die jeweilige Anwendung haben, oder man müsste den risikoreichen Weg des Exports einzelner Seiten durchführen. Hierbei stellt sich spätestens bei zentral definierten Eigenschaften wie Wertelisten/LOV‘s, Authentifizierung, Autorisierung und vielen weiteren Gemeinsamen Objekten die Frage, bis zu welchem Punkt man den Export einzelner Seiten gefahrlos durchführen kann.
Daher sind wir den Weg gegangen, dass eine parallele Entwicklung an APEX spezifischen Objekten und der Anwendung immer zusammen passen muss und entsprechend das Objekt in der Versionierung keinem Branch unterzogen wird, sondern ein Lock bekommt und zum späteren Zeitpunkt für die jeweilige Version des Objekt ein „Tag“ vergeben wird um dieses für eine Version zu definieren.

Zusätzlich kann man APEX Anwendungen modular entwickeln und eine Anwendung in mehrere APEX Anwendungen. Dies erhöht die Wartbarkeit und Flexibilität und verringert die Fehleranfälligkeit.

Auch sollte man innerhalb von Subversion mit Properties bzw. Keywords wie „$author$“, „$date$“ , „$revision$“ arbeiten um in jedem Objekt die Informationen über Änderungsdatum, Person und Version mitzutragen. Die Revision eines eingecheckten Objekts kann man im Ticketsystem dann mit ablegen.

Innerhalb der Versionierung muss man im Vergleich zu vielen anderen Programmiersprachen gerade im Bereich der Datenbank genau überlegen, wie man inkrementelle Updates anlegt und wie man eine Vollständige Auslieferung unterstützt. Da Tabellen, die bereits Inhalte beinhalten nicht in jeder Art aktualisiert werden können, müssen bei einem inkrementellen Update komplexe Aktualisierungsscripte geschrieben werden um Daten zu migrieren und die Tabellenstruktur anzupassen.
Dafür sollte zu jedem Objekt was in der Datenbank diesen Einschränkungen unterliegt (Tabellen, Constraints, Indices) sowohl mit einem Erstellungsscript komplett erstellt werden können, aber zusätzlich mit einem „Alter“-Script jeweils angepasst werden. Dieses „Alter“-Script sollte die Version des Datenbankmodells beachten und nur zur jeweils passenden Vorversion lauffähig sein.
Ggf. kann man innerhalb der Versionierung Skripte bereitstellen, die Abhängig von den Keywords und der jeweiligen Version automatisiert ein Objekt „taggen“ bzw. aus der aktuellen Version entfernen.
Basierend auf dem Inhalt der jeweiligen Version unterhalb des „/tags“ Ordners im Bereich Subversion kann man dann alle Objekte installieren.

Weiter sollte man beachten, dass man innerhalb der Datenbank unter den verschiedenen Skripten auch eine Reihenfolge definieren muss. Dazu kann man jedes Objekt mit Hilfe des Keywords der ID identifizieren und ein Keyword der $PREDECESSOR_ID$ bereitstellen.

Eine weitere Möglichkeit die Objekte einer Auslieferung zu definieren ist eine kleine APEX Anwendung zur Hinterlegung der jeweiligen Objekte und des Pfades in Subversion, sowie der Vorgängerbeziehungen. Der Vorteil dieser Mechanik ist die bessere visuelle Darstellung der Abhängigkeiten und die Absicherung, dass Objekte die in Subversion eingecheckt sind auch an der korrekten Stelle liegen.

Die APEX Besonderheiten

Um innerhalb der Entwicklung in der APEX Anwendung arbeiten zu können ist es von großer Bedeutung dass auch hier ein Locking stattfindet. Man sollte dabei beachten, dass das Locking bisher leider nur auf Seitenebene möglich ist. Objekte innerhalb der „Gemeinsame Objekte“ können leider nicht gesperrt werden.
Weiter muss man bei der Sperrung einer Seite beachten, dass APEX erst in einer Version ab 4.X den Fehler behoben hat, dass man trotz Sperrung einer Seite beim Durchblättern durch Items/Validierungen oder Spalten im Report trotzdem eine Änderung durchgeführt hat.

Auch sollte man bei der APEX Entwicklung darauf achten, dass man bei der Bearbeitung einer Region, Item, Validierung oder Prozess nicht in einem weiteren Tab die Seite wechselt und danach auf „Apply Changes“ drückt. Denn dann befindet sich das bearbeitende Objekt auf der anderen Seite wieder.

Fazit
In größeren APEX Projekten sind Faktoren wie Transparenz, Wissen, Vorgaben und die Vorbereitung enorm wichtig. Ein großes Projekt kann selbst bei einem guten Fachkonzept bei Mangel an Transparenz und Wissensvermittlung scheitern. Während in kleinen Projekten alle Probleme, Fehler und Sonderfälle beim Entwickler bekannt sind, so müssen diese in einem Großprojekt transparent und umfassend festgehalten werden. Jede technische Besonderheit muss zusätzlich mit dem Architekten abgestimmt werden, damit Seiteneffekte und unterschiedliche Funktionsweisen der Anwendung minimiert werden.
APEX eignet sich sehr gut für größere Projekte, nur ist es ausschlaggebend, dass man die in diesem Vortrag aufgeführten Besonderheiten beachtet und ein großes Projekt entsprechend angeht.

Kontaktadresse:

Name
MT AG
Balcke-Dürr-Allee, 9
D-40882 Ratingen

Telefon:	+49 (0) 2102 309 61-0
Fax:	+49 (0) 2102 309 61-101
E-Mail	oliver.lemm@mt-ag.com
Internet:	www.mt-ag.com